

**VŠB -Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra telekomunikační techniky**

## **Virtuální Datalogger na bázi NI cRIO**

**Virtual Datalogger based on NI cRIO**

2011

Bc. Martin Manek

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra telekomunikační techniky

## Zadání diplomové práce

Student: **Bc. Martin Manek**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: Virtuální Datalogger na bázi NI cRIO  
Virtual Datalogger based on NI cRIO

Zásady pro vypracování:

1. Základní principy virtuální instrumentace, hodnocení jejích výhod.
2. Zvládnutí technologie virtuální instrumentace na bázi programovatelných polí.
3. Definice funkcí virtuálního Dataloggeru - snímání a zobrazení signálů, funkce jejich analýzy, ukládání dat k jejich dalšímu zpracování a jejich následná vizualizace.
4. SW návrh aplikace Dataloggeru na bázi programovatelných polí.
5. Zhodnocení navržených řešení, zkušenosti z měření.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.


Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Ludvík Koval, Ph.D.**

Datum zadání: 19.11.2010

Datum odevzdání: 06.05.2011



  
prof. RNDr. Vladimír Vašínek, CSc.  
vedoucí katedry

  
prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

---

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 6. května 2011

.....  
Bc. Martin Manek

## Abstrakt

Diplomová práce se zabývá problematikou snímání elektrických signálů a jejich archivace s možností zpětného zobrazení. Datalogger je konstruován na principu virtuální instrumentace, využívá osobní počítač a vestavěný systém CompactRIO s FPGA průmyslovými moduly I/O. CompactRio představuje rozšiřitelnou hardwarovou architekturu, spojující kontrolér s procesorem a operačním systémem reálného času, rekonfigurovatelné hradlové pole FPGA a vstupně výstupní moduly v jediném šasi. Vše vytvořené firmou National Instruments. Digitalizované vstupní signály vizualizuje na obrazovce monitoru a ukládá do souborů na disk. Také nabízí vizualizaci starších dat uložených v souborech. Vstupní signály mohou být analogové i digitální. Digitalizovaná data lze zpracovávat základními postupy jako je filtrace, je možno provést funkce součtu, inverze, derivace integrace a FFT analýzu.

## Klíčová slova

Virtuální přístroj, datalogger, CompactRIO, FPGA, Labview, grafické programování, FFT.

## Abstract

The dissertation deals with problems of scanning electrical signals, their archivation and possibility of visual reproduction. Datalogger is designed as virtual instrument. Instrument is based on PC and embedded systém CompactRIO with FPGA industry modules I/O. CompactRIO hardware is open hardware architecture, connecting controller to processor and RTOS, programmed FPGA logic array and I/O modules in one chassis. All created by National Instruments company. Input signals are digitized and viewed on PC monitor. There is also possibility of data saving into hard disc files. We can view and analyze older stored data too. Inputs are analog and digital signals. Digitized data can be processed by filters. We can also apply other function as add, inverse, derivate, integrate, FFT analysis.

## Keywords

Virtual instrument, datalogger, CompactRIO, FPGA, Labview, graphical programming, FTT.

## Seznam použitých symbolů a zkratek:

<i>A</i>	- amplituda
<i>A/D</i>	- analogově – digitální
<i>AI</i>	- umělá inteligence
<i>AMUX</i>	- analogový multiplexer
<i>A<sub>0</sub></i>	- střední hodnota
<i>CD</i>	- kompaktní disk
<i>CPU</i>	- centrální procesorová jednotka
<i>cRIO</i>	- CompactRIO
<i>CompactRIO</i>	- Compact Realtime Input/Output. Hardwarová platforma od firmy National Instruments určená pro aplikace bezobslužného měření a řízení. cRIO systémy mají integrovanou hardwarovou architekturu, která spojuje kontrolér s procesorem a operačním systémem reálného času, rekonfigurovatelné hradlové pole FPGA.
<i>D/A</i>	- digitálně (číslicově) – analogový
<i>DAP</i>	- digitálně analogový převodník
<i>DC</i>	- stejnosměrný proud (Direct Current)
<i>DIO</i>	- digitální vstup/výstup (Digital Input/Output)
<i>DMA</i>	- metoda používaná pro přímý přístup do operační paměti počítače bez účasti procesoru (Direct Memory Access)
<i>EEPROM</i>	- Electrically erasable programmable read-only memory
<i>EISA</i>	- standardní sběrnice PC (Extended Industrial Standard Architecture)
<i>FIFO</i>	- paměť typu First In First Out. Princip zpracování dat „první dovnitř, první ven“.
<i>FPGA</i>	- field-programmable gate array (terénně-programovatelné hradlové rozhraní)

<i>GND</i>	- uzemnění ( <i>Ground</i> )
<i>GPIB</i>	- standardní sběrnice používaná pro řízení měřicích přístrojů počítačem ( <i>General Purpose Interface Bus</i> )
<i>GUI</i>	- grafické uživatelské rozhraní ( <i>Graphical User Interface</i> )
<i>HP VEE</i>	- grafické vývojové prostředí ( <i>Hewlett Packard Visual/Engineering Environment</i> )
<i>I/O</i>	- vstup/výstup ( <i>Input/Output</i> )
<i>IPCI</i>	- průmyslová sběrnice ( <i>Industrial Peripheral Component Interconnect</i> )
<i>ISA</i>	- standardní sběrnice PC ( <i>Industrial Standard Architecture</i> )
<i>LabVIEW</i>	- grafické vývojové prostředí ( <i>Laboratory Virtual Instruments Engineering Workbench</i> ) od firmy <i>National Instruments</i>
<i>MUX</i>	- multiplexer
<i>n</i>	- počet bitů A/D převodníku
<i>NI</i>	- <i>National Instruments</i>
<i>PC</i>	- osobní počítač ( <i>Personal Computer</i> )
<i>PCI</i>	- procesorově nezávislá sběrnice ( <i>Peripheral Component Interconnect</i> )
<i>RA</i>	- relativní přesnost
<i>RS 232</i>	- označení sériového rozhraní v počítači
<i>S/H</i>	- vzorkovací obvod ( <i>Sample/Hold</i> )
<i>Sub-VI</i>	- podmnožina virtuálního přístroje nebo jeho část
<i>t</i>	- čas [s]
<i>TTL</i>	- tranzistorově orientovaná logika ( <i>Transistor Transistor Logic</i> )
<i>USB</i>	- univerzální sériová sběrnice ( <i>Universal Serial Bus</i> )
<i><math>\nu_f</math></i>	- vysokofrekvenční
<i>VL-bus</i>	- sběrnice PC ( <i>Video Local Bus</i> )

<i>VXI</i>	- <i>průmyslový standard pro budování měřicích systémů pro automatizovaný sběr dat (VME eXtension for Instrumentation)</i>
<i>Zap./Vyp.</i>	- <i>Zapnout/Vypnout</i>



## Obsah

<b>1</b>	<b>Úvod.....</b>	<b>1</b>
<b>2</b>	<b>Základní principy virtuální instrumentace.....</b>	<b>2</b>
2.1	Technologie virtuální instrumentace .....	2
2.2	Měřicí přístroje z období klasické instrumentace .....	3
2.3	Měřicí přístroje s komunikačním rozhraním .....	3
2.3.1	Výhody měřicích přístrojů vybavených komunikačním rozhraním.....	3
2.3.2	Porovnání měřicích přístrojů.....	5
2.4	Virtuální měřicí přístroje .....	5
2.5	Virtuální instrumentace .....	6
2.6	Vývojová prostředí Virtuální instrumentace.....	7
2.6.1	Vývojová prostředí firmy National Instruments .....	7
2.6.2	Vývojová prostředí ostatních výrobců. ....	14
<b>3</b>	<b>Definice funkcí Dataloggeru .....</b>	<b>17</b>
3.1	Definice Dataloggeru.....	17
3.2	Základní rozdělení Dataloggerů .....	17
3.3	Definice funkcí virtuálního dataloggeru .....	18
3.3.1	Snímání a zobrazování signálů .....	18
3.3.2	Ukládání snímaných signálů .....	18
3.3.3	Doplňkové funkce digitálního dataloggeru.....	18
<b>4</b>	<b>Programovatelné hradlové pole a Platforma CompactRIO .....</b>	<b>19</b>
4.1	Programovatelné hradlové pole .....	19
4.1.1	Základní vlastnosti Programovatelného hradlového pole .....	19
4.1.2	Architektura FPGA .....	20
4.1.3	Programování hradlového pole .....	21

4.2	Platforma CompactRIO firmy National Instruments .....	21
4.2.1	Charakteristické rysy .....	22
4.2.2	Parametry .....	22
4.2.3	Možnosti použití .....	23
4.2.4	Kontrolér .....	23
4.2.5	FPGA (šasi).....	23
4.2.6	I/O moduly .....	24
4.2.7	Vývoj aplikací pro platformu CompactRIO.....	25
<b>5</b>	<b>Realizace hardwarové části virtuálního dataloggeru .....</b>	<b>27</b>
5.1	Technické parametry využitých hardwarových komponentů .....	28
5.1.1	Parametry kontroléru NI cRIO – 9073 .....	28
5.1.2	Parametry C modulů NI 9215, NI 9422 .....	29
5.1.3	Osobní počítač Lenovo T 61 .....	31
<b>6</b>	<b>Realizace softwarové části virtuálního dataloggeru .....</b>	<b>32</b>
6.1.1	Komunikace na základě příkazu .....	33
6.1.2	Návrh aplikace z pohledu uživatele .....	33
6.2	Aplikace FPGA.....	34
6.3	Aplikace RT .....	36
6.3.1	Cyklus komunikace s FPGA .....	37
6.3.2	Cykly komunikace s Host .....	38
6.4	Aplikace HOST .....	40
6.4.1	Prvky horního panelu .....	40
6.4.2	Prvky tabelového okna.....	41
6.4.3	Programová struktura aplikace HOST .....	43
6.5	Aplikace Vizualizace .....	45

6.5.1	Graf .....	45
6.5.2	Ovládací prvky grafu .....	45
6.5.3	Prvky aplikace Vizualizace .....	46
6.6	Manipulace s grafy .....	47
6.6.1	Manipulace s průběhem grafu .....	47
6.6.2	Legenda grafu a formátování grafu .....	47
6.7	Komunikace přes TCP/IP pomocí STM .....	49
6.7.1	Implementace STM protokolu v LabVIEW .....	49
6.8	Použitý souborový systém .....	50
6.8.1	Souborový systém TDMS .....	50
6.8.2	Formát souborového systému TDMS .....	50
<b>7</b>	<b>Závěr .....</b>	<b>52</b>
	<b>Použitá literatura .....</b>	<b>54</b>
	<b>Seznam příloh .....</b>	<b>55</b>

# 1 Úvod

V současnosti pomáhá vědě a technice v pokroku instrumentace. Vědci a inženýři používají instrumentace k měření a řízení ve všech oborech lidské činnosti, kvalita našeho života, např. urychlení výzkumu ve zdravotnictví, závisí na jejím budoucím vývoji. Současným trendem je přechod od klasické instrumentace k virtuální. Zástupce klasické instrumentace, digitální voltmetr, je zaměřený pouze na měření napětí, jeho ovládání a vzhled jsou tomu přizpůsobeny. U klasické instrumentace je majoritní hardwarová stránka. Naopak ve virtuální instrumentaci je nejvyšší důležitost vkládána do vývoje softwaru. Virtuální přístroj založený na moderní výpočetní technice je vysoce flexibilní a univerzální. Pro komerční aplikace je důležitou vlastností rychle vyvinout prototyp zařízení a v co nejkratší době otestovat reakci trhu (spirálový model vývoje softwaru), což virtuální instrumentace umožňuje.

Datalogger (neboli záznamník dat) je elektronické zařízení určené pro sběr a uchování naměřených údajů. Lze ho použít v širokém spektru oborů, uchovávaná data lze využít k analýze, vyvození závěrů nebo objasnění příčin. Nejznámější datalogger ve verzi “černá skříňka” se používá zpravidla v letecké a jiné dopravě za účelem zaznamenávání nejdůležitějších parametrů letu pro objasnění příčin případné havárie.

Cílem mé diplomové práce je navrhnout a naprogramovat datalogger na bázi technologie NI CompactRIO pomocí personálního počítače, s užitím softwarového systému pro automatizaci a měření LabVIEW a příslušného hardwaru. Datalogger by měl být snadno ovladatelný, přičemž by měl splňovat zadané funkce. Program plně využije možnosti programového systému LabVIEW. Naměřený signál bude možné uložit v \*.tdms formátu na disk a také načíst signál uložený na disku. Snahou je dosáhnout vysoké přesnosti přístroje s přihlédnutím k možnostem použitého hardwaru.

V teoretické části se budu zabývat významem virtuální instrumentace, jejími výhodami a nevýhodami a stručným popisem vývojových prostředí, dále problematikou dataloggerů, technologii cRIO a programovatelných hradlových polí. Zaměřím se především na popis zařízení, která v praktické části využívám pro zkonstruování virtuálního dataloggeru.

## 2 Základní principy virtuální instrumentace

### 2.1 Technologie virtuální instrumentace

Pojem Virtuální instrumentace popisuje kombinaci programovatelných přístrojů a všeobecně použitelného osobního počítače. Můžeme se tedy na ni dívat jako na spojení (kombinaci) softwaru a hardwaru za účelem analýzy a vyhodnocování rozličných elektrických veličin. Softwarové komponenty této koncepce umožňují uživatelům vytvářet širokou škálu měřicích, řídicích a regulačních systémů, založených na běžně vyráběných hardwarových komponentách. Tímto spojením pak vznikají velmi přesné modulární systémy.

Virtuální instrumentace ve svém obsahu skrývá kombinaci základního hardwarového vybavení, nejčastěji v podobě snímací karty s analogově číslicovým převodníkem, programové nástavby s uživatelem definovanými funkcemi aplikace a všeobecně použitelného osobního počítače. Vše je těsně integrováno tak, aby se podstatně rozšířil okruh lidí schopných obsluhovat a navrhovat virtuální zařízení.

Předností virtuální instrumentace je dokonalá aplikace s vlastnostmi přesně určenými uživatelem, která není omezena na pouze výrobcem implementované funkce profesionálního přístroje, ale počítá s doplňováním dalších užitečných schopností, případně s modifikací těch stávajících.

Virtuální instrumentace samozřejmě vychází z klasické analogové technologie, představované přístroji s ručním ovládáním pomocí prvků na čelním panelu, kterou podstatně rozšiřuje o možnosti záznamů výsledků a jejich dalšího zpracování. Navíc přidává takové možnosti, jakými jsou úpravy uživatelského rozhraní, měřicích schopností nebo výpočetních vlastností měřicího zařízení.

Dnešní měřicí systémy jsou tvořeny spojením univerzálních počítačů s přístroji jak nové generace (např. zásuvné měřicí karty), tak i těmi tradičními (klasické měřicí přístroje s potřebným rozhraním, např. RS232, nebo lépe GPIB). Tyto systémy mohou dosáhnout vyšší úrovně výkonu a flexibility kombinováním nových ovládacích programů s nejrůznějšími třídami měřicích přístrojů. S technikami, jako jsou orientované programování a sdílení paměti PC, je možné maximálně využít potenciál moderních počítačů při sběru, analýze a zobrazení měřených dat.



Obr. 1: Vývoj měřicí techniky v čase z hlediska narůstající flexibility

## 2.2 Měřicí přístroje z období klasické instrumentace

U starších analogových měřicích přístrojů bez komunikačního rozhraní (voltmetry, ampérmetry, multimetry, generátory, osciloskopy, ...) definuje funkce měřicího přístroje jednoznačně jeho výrobce a koncový uživatel je omezen pouze na funkce, jež měl od výrobce v přístroji implementovány - flexibilita měřicího systému je nízká. Narazí-li potřeba uživatele na funkci, kterou přístroj neobsahuje, zůstává jeho potřeba neuspokojena.

V této fázi vývoje měřicí techniky je i způsob přebírání naměřených hodnot koncovým uživatelem obvykle omezen na opisování naměřených dat z displeje měřicího přístroje, kde může nastat chybné odečtení naměřených hodnot, které je dalším potenciálním zdrojem chyb měření.

Zejména posledně zmiňované aspekty klasické instrumentace vedly ke zlepšení efektivnosti konfigurace měřicího přístroje a přebírání naměřených hodnot s vyloučením uživatele přímou komunikací měřicího přístroje s počítačem. Napomáhá tomu bouřlivý vývoj a otevřená architektura personálních počítačů.

## 2.3 Měřicí přístroje s komunikačním rozhraním

### 2.3.1 Výhody měřicích přístrojů vybavených komunikačním rozhraním

Propojení měřicího přístroje s počítačem nám umožňuje doplnit funkce, implementované do jeho firmware výrobcem, funkcemi, které lze naprogramovat přes připojený personální počítač. V tomto případě lze dokonce omezit funkce měřicího přístroje pouze na snímání měřených dat a jejich přenos do počítače přes komunikační rozhraní. Dále proběhne v personálním počítači

jejich následné zpracování, při kterém lze softwarově realizovat i funkce, které měřicí přístroj od výrobce neobsahuje.

Takto lze udělat např. z digitálního osciloskopu jednoduchým způsobem frekvenční analyzátor, který je obvykle několikanásobně dražší (osciloskop pouze sejme měřený průběh a pošle jej přes komunikační rozhraní do počítače, kde je možné podrobit tato data frekvenční analýze např. pomocí FFT).

Flexibilita celého měřicího systému se oproti klasickému měřicímu přístroji bez komunikačního rozhraní výrazně zvýší. Potřebuje-li uživatel funkci, kterou přístroj nemá implementovanu výrobcem, může ji následně realizovat softwarovými prostředky na připojeném počítači.

Vývojáři měřicí i výpočetní techniky dnes používají stejných návrhů a komponentů při návrhu zapojení vyvíjených zařízení (využití mikroprocesorů, sběrnic, pamětí, ...). Vyspělejší moderní měřicí přístroj je obvykle multiprocesorový systém, který je vybaven základní vrstvou softwaru. Tento software řídí jeho činnost a nazývá se firmware.

Tato koncepce umožňuje měnit chování přístroje v určitých omezených mezích a umožňuje použití ovládacích prvků na čelním panelu, které jsou volně programovatelné. Přibývající počet funkcí, které výrobci měřicích přístrojů implementují do svých výrobků, by při klasické koncepci (jedna funkce – jeden ovládací prvek) vedla k velmi nepřehledným a rozměrným čelním panelům přístrojů. Proto se dnes více používá umístění ovládacích tlačítek na okraj displeje, na kterém je v každé fázi konfigurace daného přístroje přiřazena konkrétnímu tlačítku jiná funkce popsaná proměnným nápisem na displeji.

Při rozboru propojení personálního počítače a měřicího přístroje přes komunikační rozhraní přišli výrobci měřicí techniky na to, že se z měřicího přístroje někdy využije pouze vstupních obvodů a paměti dat, což umožnilo další krok vývoje této techniky směrem k tzv. virtuálním měřicím přístrojům.

### 2.3.2 Porovnání měřicích přístrojů

Srovnání klasických měřicích přístrojů a virtuálních měřicích přístrojů je uvedeno v tabulce 1.

Tabulka 1: Srovnání klasických a virtuálních měřicích přístrojů

Hledisko	Tradiční přístroj	Virtuální přístroj
<b>Funkce definuje</b>	Výrobce	uživatel
<b>Orientace přístrojů a jejich propojitelnost</b>	specifické podle funkcí, používané hlavně samostatně, s omezenou propojitelností	specifické podle aplikace, propojitelnost s návazností na počítačové sítě a periferie
<b>Klíčový komponent</b>	hardware	software
<b>Náklady na výrobu a opakovatelnost použití</b>	vysoké	nízké, opakovatelně použitelné
<b>Architektura</b>	uzavřená, pevná množina funkcí	otevřená, pružná funkčnost odrážející výkonnost výpočetní techniky
<b>Návaznost na technologický proces</b>	pomalá – doba obměny 5 až 10 let	rychlá – doba obměny 1 až 2 roky
<b>Náklady na vývoj a údržbu</b>	vysoké, horší poměr výkon/cena	nízké, dobrý poměr výkon/cena

## 2.4 Virtuální měřicí přístroje

Filosofie virtuálních měřicích přístrojů má velkou budoucnost a umožňuje při stejných výkonnostních parametrech klasické měřicí techniky vytvářet přístroje, jejichž funkce plně odpovídají požadavkům uživatele, protože jsou softwarově realizovány. Dále je zde možné doplňovat další funkce podle měnících se a narůstajících potřeb koncového uživatele. Cena takto vytvářených virtuálních měřicích přístrojů bývá řádově nižší, než je cena klasické analogové měřicí techniky. Navíc je možno kdykoliv modifikací programu upravit vlastnosti stávajícího měřicího přístroje nebo popřípadě vytvořit jiný měřicí přístroj.

V dnešní době se u měřicích přístrojů nejčastěji vyskytují tři typy rozhraní využívané pro komunikaci s personálním počítačem (PC) - rozhraní RS 232, USB, GPIB a Ethernet. Při propojení počítače a měřicího přístroje pomocí rozhraní nebo při tvorbě virtuálních měřicích přístrojů hraje kromě hardwarových prostředků v podobě počítače a přídatných multifunkčních



karet čím dál důležitější roli software, který se stává u měřicího systému hlavním klíčovým komponentem.

V zásadě pro tvorbu aplikací lze použít i standardních programovacích jazyků (Visual Basic, C jazyk ...) v oblasti měření, ale tvorba bude zdoluhavá a náročná na znalosti detailů komunikačních protokolů apod. Mnohem efektivněji je možno tyto aplikace vytvářet ve speciálních vývojových prostředích, které jsou pro tuto oblast vybavené nástroji urychlujícími a usnadňujícími proces tvorby aplikace.

## 2.5 Virtuální instrumentace

Hlavní úlohou virtuálního přístroje je umožnění doplnění otevřené architektury personálního počítače tím, co neobsahuje, aby mohl plnit úlohu měřicího přístroje. V oblasti hardwaru se jedná o zásuvnou multifunkční kartu (zásuvná měřicí deska) obsahující konektor pro zasunutí této karty do systémové desky personálního počítače (historicky sloty ISA, EISA, dnes se používá standardně PCI sběrnice). Zásuvná měřicí deska má za úkol převod naměřených analogových signálů na číslo a vstup binárních signálů.

V oblasti využití softwaru je to vhodný program pro počítač, pomocí kterého lze realizovat všechny funkce měřicího přístroje a plnit tak úlohu firmwaru měřicího přístroje. Vyhneme se tak i duplicitě, která se vyskytuje při propojení počítače a měřicího přístroje.

Grafické rozhraní k uživateli plní jednu z hlavních součástí aplikace představující virtuální přístroj. Je to čelní panel virtuálního přístroje, na kterém má koncový uživatel k dispozici:

- indikační prvky informující o výsledcích měření,
- ovládací prvky sloužící pro nastavení parametrů měřicího přístroje.

Technologie virtuální instrumentace se v dnešní době používá taktéž u renomovaných výrobců měřicí techniky. Tyto firmy nabízejí měřicí přístroje, jejichž hardware je postaven na technologii personálních počítačů. Po zapnutí přístroje je na jeho monitoru vidět náběh standardního operačního systému Microsoft Windows a veškerý firmware je tvořen aplikací běžící nad tímto operačním systémem.

Další využití technologie virtuální instrumentace představují přístroje, u kterých lze spustit vlastní aplikace přímo na hardwarové platformě měřicího přístroje.

Pod termínem virtuální přístroj rozumíme aplikaci vytvořenou ve vývojovém prostředí, která obsahuje grafické rozhraní k uživateli připomínající čelní panel měřicího přístroje (obsahující indikační a ovládací prvky) a využívá pro vstup měřených signálů komunikace s přístrojem připojeným přes komunikační rozhraní nebo zásuvné multifunkční karty.

Virtuální přístroj si oproti klasickému přístroji ponechává veškeré výhody personálního počítače:

- ukládání naměřených dat na standardní média (USB disk, pevný disk, DVD, CD, disketa),
- snadnou změnu funkce virtuálního přístroje modifikací vytvořené aplikace,
- jednoduché připojení periférií (modem, tiskárna),
- jednoduchá konektivita do počítačových sítí, a tím možnost dálkové ovládání přístroje nebo sdílení naměřených dat.

## 2.6 Vývojová prostředí Virtuální instrumentace

Pro tvorbu a vývoj aplikací pro osobní počítač, používaných pro průmyslové prostředí, je potřeba zvolit optimální softwarový prostředek. Lze využít např. standardních programovacích jazyků, kterými jsou Visual Basic nebo jazyk C. Ovšem použití těchto jazyků je časově náročné, je nutná detailní znalost hardwarového řešení měřicího řetězce a poměrně složité syntaxe. Efektivnější je aplikace vytvářet ve speciálních vývojových prostředích (Development Environment), která poskytují programátorovi celou řadu funkcí urychlujících a usnadňujících proces tvorby aplikace. Jedná se o otevřené systémy, kde není programátor nijak omezován, neboť funkce těchto systémů se dají jednoduchým způsobem rozšiřovat podle potřeb.

### 2.6.1 Vývojová prostředí firmy National Instruments

#### 2.6.1.1 Grafické vývojové prostředí LabVIEW

Vývojoví pracovníci firmy National Instruments chtěli dát do rukou inženýrů nástroj podobné efektivity, pružnosti a síly, jakým je tabulkový procesor v rukou finančního manažera. Myšlenka, na níž stojí efektivita vývojového prostředí LabVIEW I daného na trh pro platformu počítačů Macintosh v roce 1986, je jednoduchá a vznikla původně na půdě Texaské univerzity ve skupince nadšenců v čele se zakladatelem tohoto systému Jeffem Kodovským. Hlavním záměrem je, že tím, kdo ví, co měřit, jak analyzovat a jak prezentovat data, se stává technik, který nemusí být sám vyspělým programátorem. Své představy tedy předává programátorovi zejména v podobě blokového schématu. Programátor blokové schéma převádí do syntaxe zvoleného programovacího jazyka, což je činnost časově zdlouhavá, náročná na přesnost a nepřináší již do procesu měření většinou žádné další nové informace. Cílem grafického vývojového prostředí LabVIEW je již dále nepřevádět blokové schéma, které bylo koncovým tvarem aplikace, do textové podoby.

LabVIEW (Laboratory Virtual Instruments Engineering Workbench) je grafickým vývojovým prostředím vybaveným bohatými knihovnami pro tvorbu aplikací zaměřených do oblasti měření ve všech fázích tohoto procesu - tj. sběru, analýzy a prezentace naměřených hodnot. Podporuje čtyři základní způsoby přijímání dat do počítače (z měřicích přístrojů přes komunikační rozhraní RS 232 nebo GPIB, ze zásuvných multifunkčních karet a ze systému s VXI sběrnici). Dává uživateli plnohodnotný programovací jazyk se všemi odpovídajícími

programovými a datovými strukturami v grafické podobě - tzv. G jazyk (Graphical language). Pro operační systém Windows prošlo vývojové prostředí postupně verzemi 2.5 až po aktuální verzi Labview 2010 (stav k dubnu 2011). Při realizaci praktické části diplomové práce jsem využil verzi Labview 2009.

LabVIEW je tedy grafickým vývojovým prostředím srovnatelným např. s programovacím jazykem C, ale narozdíl od něj je orientován graficky a ne textově. Výsledný produkt naprogramovaný pomocí tohoto grafického vývojového prostředí se nazývá virtuální přístroj, protože svým vzhledem, projevy a činnostmi připomíná klasický přístroj ve své fyzické podobě.

Virtuální přístroj (Virtual Instrument) vytvořený v tomto grafickém vývojovém prostředí jako základní jednotka aplikace obsahuje:

- Interaktivní grafické rozhraní (Graphical User Interface - GUI) ke koncovému uživateli  
Toto rozhraní představuje čelní panel (Front Panel) fyzického přístroje obsahující prvky pro indikaci a ovládání (LED indikátory, knoflíky, tlačítka, grafy...). Uživatel jej ovládá pomocí myši nebo klávesnice.
- Blokové schéma  
Funkce virtuálního přístroje je dána jeho blokovým schématem (Block Diagram). Blokové schéma je tvořeno ikonami představujícími v koncových blocích ovládací a indikační prvky čelního panelu a ve svých uzlových blocích jsou to bloky zpracovávající tok data. Tento blokový diagram je zdrojovou podobou každé vytvořené aplikace a nedílnou součástí virtuálního přístroje.
- Podprogramy  
Virtuální přístroj je tvořen modulární a hierarchickou strukturou. Je možné jej používat jako celý program nebo jeho jednotlivé podprogramy, které se nazývají podřízenými virtuálními přístroji (Sub-VI). Každý virtuální přístroj má svou ikonu, kterou je zastoupen v blokovém schématu, a konektor s přípojnými místy pro vstupní a výstupní signály.

Těmito charakteristickými rysy splňuje grafické vývojové prostředí LabVIEW podmínky modulárního programování. Svou aplikaci dělí uživatel na jednotlivé úlohy, pro které vytváří samostatné virtuální přístroje (subVI) a z nich buduje celou aplikaci jejich spojováním do výsledného virtuálního přístroje. Nakonec lze celou aplikaci přeložit do EXE tvaru a provozovat nezávisle na grafickém vývojovém prostředí s použitím jeho Run-Time Modulu, který je volně přístupný. Díky velké škále vyspělých ladicích prostředků a možnosti vyzkoušet funkci každého samostatného virtuálního přístroje nezávisle na jiných je ladění aplikace velmi snadné.

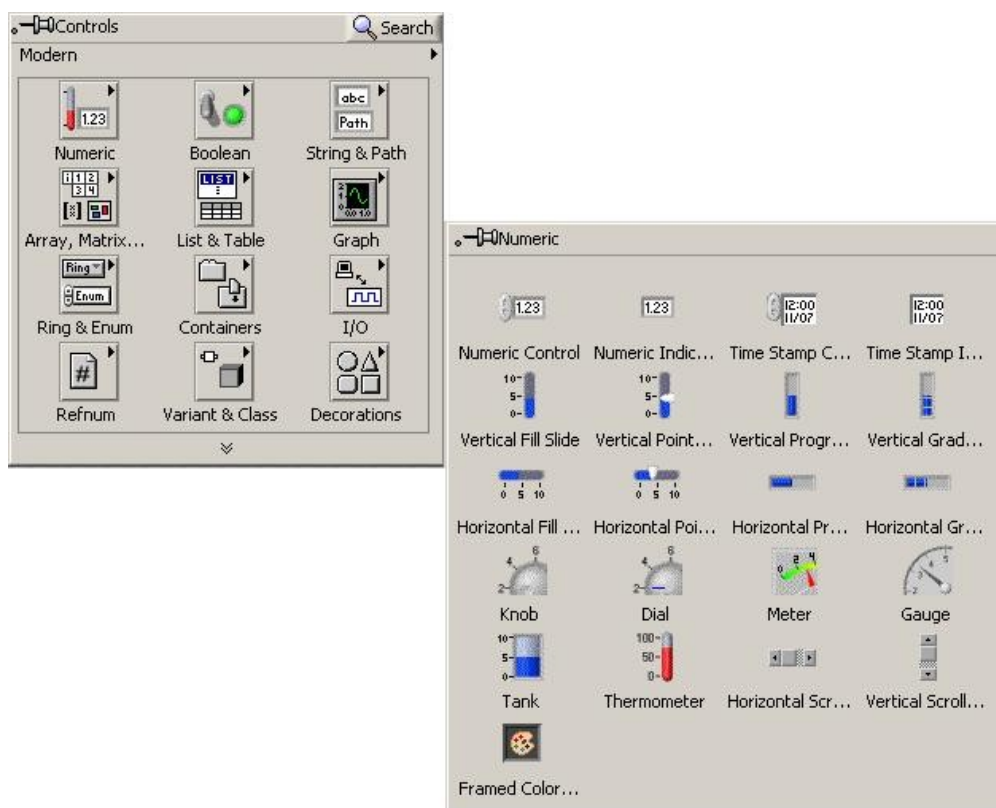
### Čelní panel

Čelní panel fyzického přístroje plní tutéž úlohu jako čelní panel grafického rozhraní virtuálního přístroje k uživateli. Pro tvorbu čelního panelu je k dispozici samostatné okno. Příklad toho, jak může vypadat čelní panel virtuálního přístroje je uveden na obrázku 1 v příloze 3.

Na čelním panelu se nachází dva typy prvků:

- Ovládací (Controls) - simulují vstupní zařízení (v blokovém schématu jsou reprezentovány vstupními bloky) a slouží k ovládání virtuálního přístroje. Zprostředkovávají vstupní informace od uživatele do aplikace, v blokovém diagramu jsou to bloky, z nichž signál vystupuje do algoritmu.
- Indikační (Indicators) - simulují výstupní zařízení (v blokovém schématu jsou reprezentovány výstupními bloky) a slouží k indikaci stavu virtuálního přístroje a výstupu výsledků. Umožňují předávání informací směrem od aplikace k uživateli, v blokovém schématu jsou to bloky, v nichž končí signálové cesty.

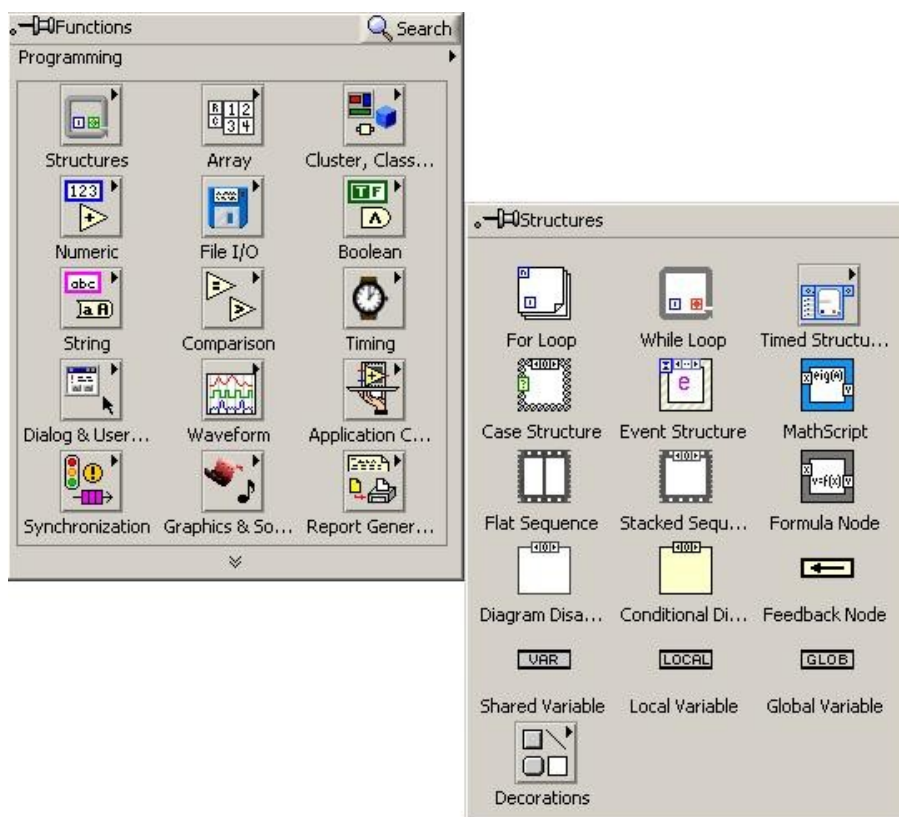
Ovládací a indikační typy prvků se na čelní panel umísťují z knihovny, která je součástí systému. Tyto objekty lze z knihovny vybírat pomocí paletových nabídek, které se zpřístupní pomocí nabídky Windows / Show Controls Palette. U vybraného objektu lze po umístění v okně čelního panelu měnit velikost, umístění, barvu atd. Kromě možnosti vyvolat tuto položku z menu existuje také rychlejší metoda, a sice vyvolat zviditelnění paletového menu stiskem pravého tlačítka myši při poloze kurzoru kdekoliv v čelním panelu. Každý objekt čelního panelu má navíc roletové menu vyvolatelné při poloze kurzoru myši na objektu pravým tlačítkem myši, kde lze měnit další vlastnosti tohoto objektu. Následující obrázek 2 ukazuje paletovou nabídku objektů čelního panelu.



Obr.2: Paletová nabídka objektů čelního panelu

### Blokové schéma

Blokové schéma je grafické vyjádření zdrojového kódu virtuálního přístroje. Vytváří se propojováním jednotlivých bloků signálovými cestami. Bloky jsou tvořeny koncovými bloky (zdrojovými a cílovými) a uzlovými bloky. Koncové bloky jsou na panel blokového schématu umístěny automaticky při konstrukci čelního panelu a uzlové bloky reprezentují bloky zpracování signálu, které lze vybírat z paletového menu Functions představujícího knihovnu funkcí v okně blokového diagramu. Paletové menu s funkcemi lze vyvolat stejným způsobem jako paletové menu objektů čelního panelu, a to přes menu Windows / Show Functions Palette nebo kliknutím na pravé tlačítko myši v okně blokového diagramu. Následující obrázek 3 ukazuje paletovou nabídku funkcí blokového diagramu.



Obr. 3: Paletová nabídka funkcí v blokovém diagramu

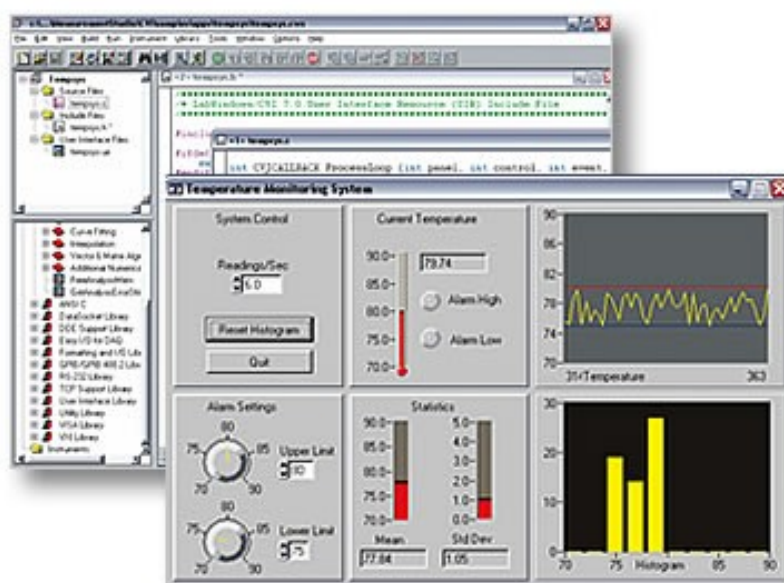
Uzlové bloky jsou podobné příkazům, operátorům, funkcím a podprogramům klasických programovacích jazyků. Bloky se propojují signálovými cestami (dráty). Dále existují tzv. struktury, nahrazující v grafické podobě standardní konstrukce používané v programovacích jazycích (podmíněný příkaz, přepínač, cyklus...). LabVIEW má i propojení na textově orientované výrazy a na vnější bloky textově orientovaného kódu.

Zdrojové a cílové koncové bloky se spojují signálovými cestami. Je možné spojit jeden zdrojový koncový blok s několika cílovými bloky, ale nelze propojit vzájemně dva zdrojové koncové bloky. Tvar a barva čáry představují definovanou signálovou cestu, dále rozlišují typ proměnné procházející daným místem.

Průběh aplikace v LabVIEW je řízen tokem dat. Tok dat (data flow) je charakterizován tím, že uzlový blok zahájí zpracování dat, má-li na všech svých vstupech k dispozici platná data, a po zpracování je posílá ke všem svým výstupům. Jednotlivé bloky mohou pracovat i současně a tím se tento způsob jednoznačně liší od sekvenčního způsobu zpracování dat v textově orientovaných jazycích, kde je běh aplikace řízen posloupností příkazů. Velmi jednoduché provedení paralelních větví zpracování aplikace umožňuje aplikace uvedeného pravidla.

### 2.6.1.2 LabWindows/CVI

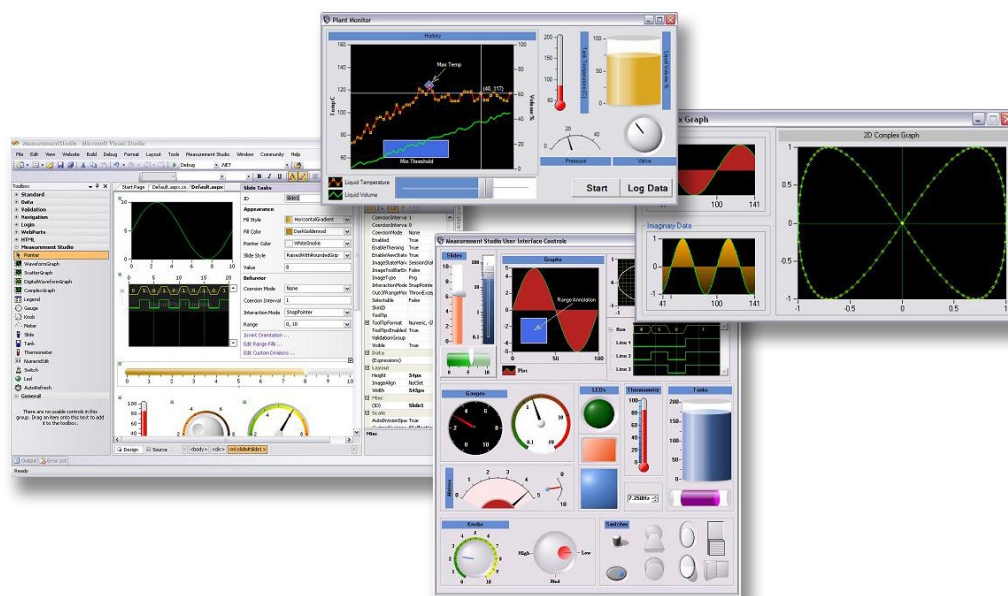
LabWindows/CVI slouží od roku 1989 k vývoji aplikací s využitím konceptu textově orientovaného programování v rozsahu ANSI C. Výhodou tohoto prostředí je kombinace grafického návrhu uživatelského rozhraní s klasickým programováním. Vlastnosti prostředí LabWindows/CVI se dají modulárně rozšířit podobně jako LabVIEW. Např. pro zpracování obrazových informací kamerových systémů se do prostředí přidává SW modul IMAQ Vision. Má zabudovanu podporu komunikačních rozhraní. Poslední verzí je LabWindows/CVI 2010 (stav k dubnu 2011).



Obr. 4: LabWindows/CVI

### 2.6.1.3 Measurement Studio

Measurement Studio bylo představeno v únoru 2000 s cílem spojit textově založené programovací prostředí, konkrétně: LabWindows/CVI, Component Works ++, Component Works. Measurement Studio představuje podporu konceptu objektově orientovaného programování v rozsahu komponentů Microsoft Visual Studio, NET 2003; 6.0 a C#.



Obr. 5: Measurement Studio

Získal ocenění E&E's v roce 2006 jako nejlepší produkt v kategorii „Design Tools and Software“. Poslední verzí je Measurement Studio 2010 (stav k dubnu 2011).

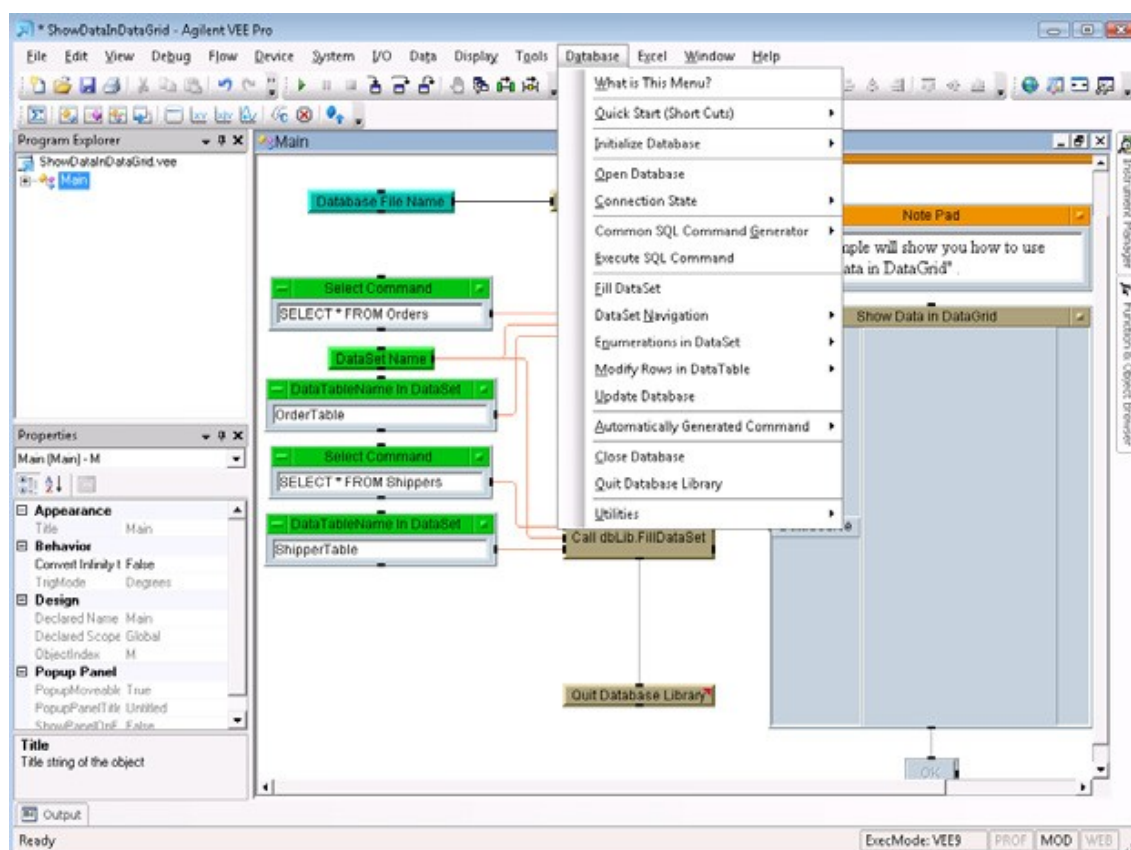


## 2.6.2 Vývojová prostředí ostatních výrobců.

### 2.6.2.1 Agilent VEE (Visual Engineering Environment)

Agilent VEE je grafické vývojové prostředí od Agilent Technologies určené pro automatizaci, měření a datovou analýzu. VEE je zkratka pro Visual Engineering Environment (Grafické Technologické Prostor) vyvinuté firmou Hewlett Packard pod označením HP VEE, později převzaté firmou Agilent Technologies a oficiálně přejmenované na Agilent Vee.

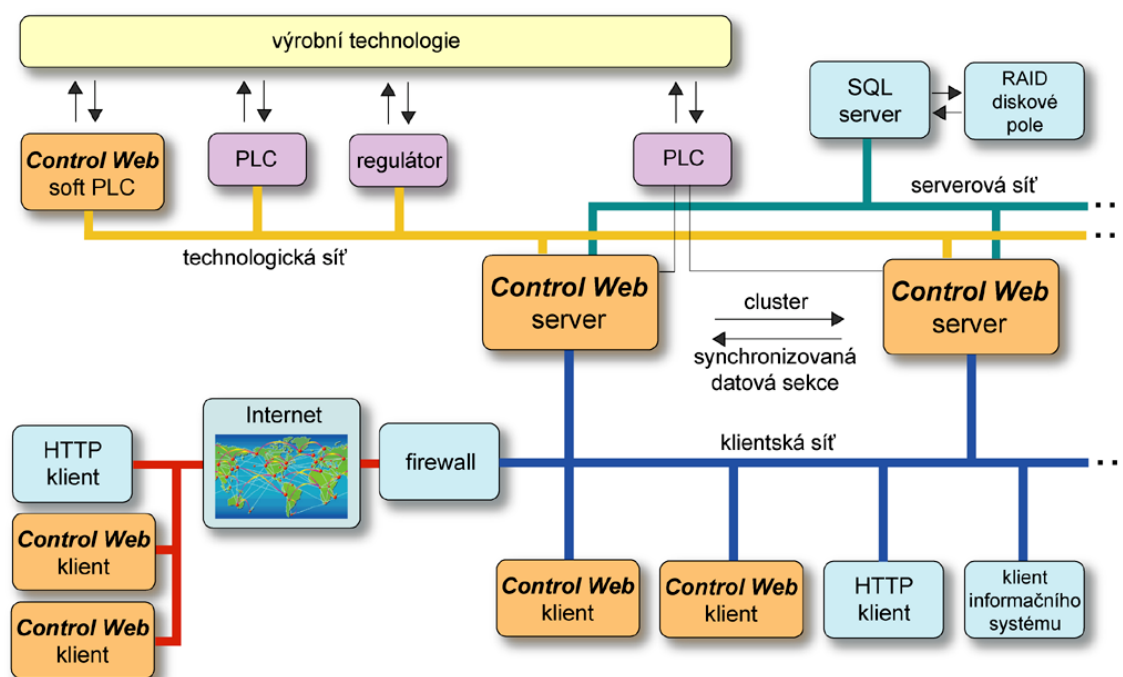
Jedná se o programovací jazyk vysoké úrovně používající grafické zobrazení podobné vývojovému diagramu. Díky otevřené struktuře software Agilent VEE pracuje bez problémů s hardwarem a softwarem od téměř všech výrobců (NI, MathWorks, MATLAB 2007, Microsoft Windows Vista, Microsoft Office 2007 atd.). V programech vytvořených ve VEE lze volat programy v C/C++ a naopak v programech vytvořených v C/C++ lze volat HP VEE např. při vytváření panelů, přístrojů či uživatelských panelů. Vytvořené programy jsou flexibilní, expandovatelné a kompatibilní s posledními světovými standardy. Poslední verzí je Agilent VEE 9.2 k dubnu 2011.



Obr. 5: Agilent VEE

### 2.6.2.2 Control Web

Je to již více než 16 let od začátku vývoje systému Control Panel (firmy Alcor Zlín), předchůdce současného prostředí Control Web vyvinutého již firmou Moravské přístroje a.s.. Control Web je programovým systémem, který dokáže vystupovat v mnoha rolích. Může pracovat v řídicích jednotkách strojů, může spojovat výrobní technologii s informačním systémem podniku, může být datovým serverem s mnoha webovými klienty, může modelovat a simulovat procesy, dokáže vytvářet náročné vizualizace. Mezi nejzajímavější vlastnosti Control Web patří třírozměrný vykreslovací systém 3D. Tato technologie počítačové grafiky může podstatně zvýšit rychlost, působivost i přehlednost vizualizací. Control Web je přístupný nástroj, který umožní levně realizovat řízení např. malé vodní elektrárny, ale také je to prostředek tvorby rozsáhle distribuované průmyslové aplikace s desítkami tisíc měřených bodů, obsahující stovky operátorských obrazovek a pracující na řadě počítačů zapojených do sítě. Umí také pracovat jako programový most mezi SQL databází, WWW prohlížeči a GSM sítí. Poslední verzí je Control Web 6.0 k dubnu 2011.



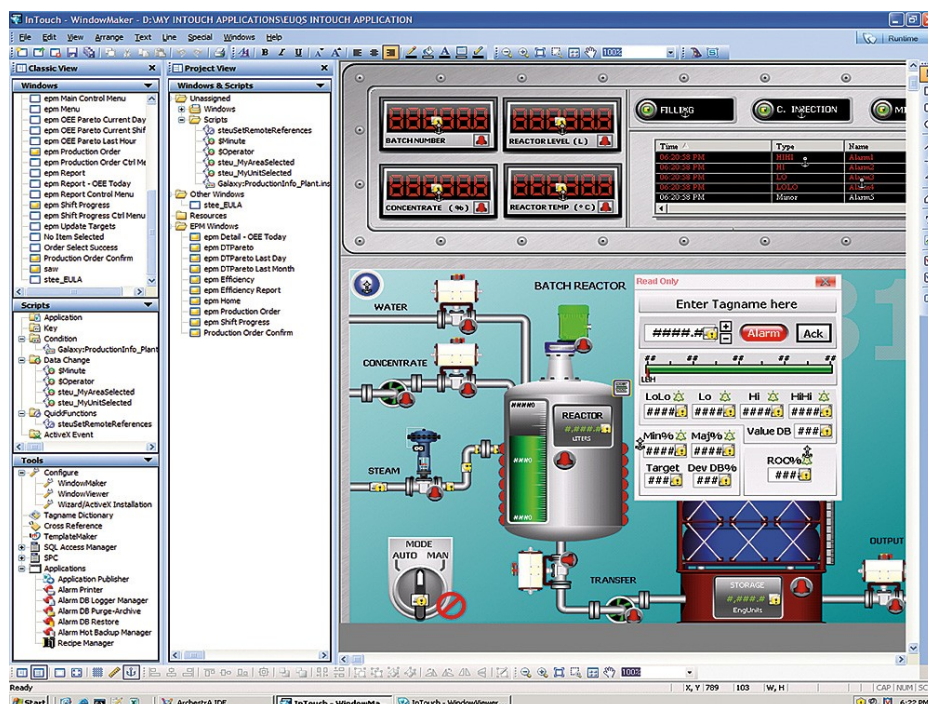
Obr. 5: Prostředí Control Web jako integrační prvek v systémech průmyslové automatizace

### 2.6.2.3 Wonderware Intouch

System pro vizualizaci, sběr dat a „supervizní“ řízení technologických procesů kategorie SCADA/HMI s uživatelskou přístupností a celosvětovým rozšířením. Umožňuje snadno vytvořit grafické zobrazení jakýchkoliv výrobních technologií na monitoru počítače, jejich ovládání a dynamické animace, které názorně v reálném čase zobrazují aktuální stavy provozovaných technologických systémů ve výrobě. InTouch slouží rovněž jako jednotné vizualizační rozhraní (front-end) pro další produkty softwarové platformy Wonderware System Platform“ obsahující základní serverové produkty (back-end):

- Wonderware Application Server (průmyslový aplikační server),
- Wonderware Historian Server (databáze pro sběr, ukládání a poskytování velkého množství technologických dat v reálném čase),
- Wonderware Information Server (výrobní informační portál pro publikaci výrobních dat do prostředí Internet/intranet),
- Wonderware Komunikační servery (komunikační programy pro výměnu dat s řídicími systémy (PLC, DCS, I/O atd.) všech významných výrobců (Siemens, Schneider, Rockwell, GE Fanuc, ZAT atd.). Podporováno je více než 600 typů komunikačních rozhraní.

Poslední verzi je *InTouch 10.1 HMI Software* k dubnu 2011.



Obr. 5: Wonderware Intouch

## 3 Definice funkcí Dataloggeru

### 3.1 Definice Dataloggeru

Datalogger je hardwarový nebo softwarový přístroj určený k ukládání dat. To zahrnuje mnoho systémů sběru dat, jako jsou zásuvné karty do PC nebo systémy se sériovou komunikací používající počítač k záznamu dat v reálném čase. Avšak většina výrobců považuje za datalogger samostatný přístroj, který umí číst různé typy elektrických signálů a uchovává data ve vnitřní paměti pro jejich pozdější přenos do počítače. Jako příklad dataloggeru lze uvést černou skříňku užívanou v dnešním letectví. Jedná se o „multidatalogger“, který pro případ nehody uchovává hodnoty přístrojů letadla, rádiovou komunikaci pilota i komunikaci posádky v kokpitu.

### 3.2 Základní rozdělení Dataloggerů

Dataloggery se dají rozdělit na dva základní druhy: hardwarový a virtuální (softwarový). Zpravidla oba obsahují alespoň část od hardwaru i softwaru, ať už se jedná o zdrojový řídicí kód u hardwarového, nebo měřicí kartu u softwarového. Výhodou hardwarových dataloggerů je jejich samostatná činnost nezávislá na počítači, avšak pro další zpracovávání nebo vizualizaci je již ve většině případů potřeba osobní počítač. Zvláštní skupinou mohou být hardwarové dataloggery opatřené např. LCD displejem. Softwarové, nebo lépe řečeno virtuální dataloggery jsou koncipovány jako personální počítač s měřicí kartou.

Dále můžeme dataloggery dělit podle:

- **vstupního signálu:** Jedná se o druh fyzikální veličiny přiváděné na vstup. Některé dataloggery pracují jen s určitým vstupem, jiné jsou programovatelné pro různé typy vstupů. Veličinami mohou být:

st. napětí/proud	světlo zap./vyp.	nárazy/zrychlení
mústek/pnutí/síla/tlak	motor zap./vyp.	zvuk
rosný bod	pH a tlak	teplota
události/stavy	procesní signály	termistory
frekvence	relativní vlhkost	termočlánky
hladina	odporové teploměry	analog / digit

- **počtu vstupů :** Dataloggery mohou být jak s jedním vstupem, tak i s mnoha vstupy. Některé dataloggery zpracovávají až stovky vstupů.
- **rychlosti a paměti:** V porovnání se systémem sběru dat v reálném čase mají dataloggery nízkou vzorkovací frekvenci. Je to pochopitelné, protože ukládají data do vnitřní limitované paměti. Čím větší je rychlost dat, tím více paměti vyžaduje. Proto je při specifikaci dataloggeru důležité určit frekvenci vzorkování a čas, po který se má vzorkování provádět, a pak lze spočítat velikost požadované paměti.

- **další:** Dále můžeme dataloggery dělit podle napájení, rozměrů, atd..

### 3.3 Definice funkcí virtuálního dataloggeru

Hlavní funkcí virtuálních dataloggerů je ukládání snímaných signálů na dané medium (pevný disk). Očekávanou funkcí je vizualizace signálu během i mimo ukládání signálu, zpětná vizualizace uložených signálů, konfigurace měření. Doplnkovými funkcemi mohou být například zpracování signálu (spektrální analýza, filtrace, matematické funkce), možnost exportu dat do rozličných datových formátů, tisk zobrazeného signálu.

#### 3.3.1 Snímání a zobrazování signálů

Měřicí karta odesílá do počítače hodnoty každého vzorku signálu v dané veličině. Možnost snímání signálu je určena typem měřicí karty, která má danou maximální vzorkovací frekvenci, rozsah hodnot, nabízené triggorovací možnosti. Signál je možno snímat nepřetržitě nebo odebrat určitý počet vzorků. Hodnoty mohou být zobrazovány textově například ve formě tabulky, ve které je ke každému vzorku přiřazena jeho hodnota, nebo graficky ve formě grafu. Možností zobrazení signálu v grafu je mnoho. Pro většinu případů se volí čas pro X osu a snímanou veličinu pro Y osu. Zobrazení lze volit např. z grafů, jako jsou spojnicový, bodový, spojnicový s vyznačenými body, s vyplněním k nulové úrovni, čárový, sloupcový.

#### 3.3.2 Ukládání snímaných signálů

Jak již bylo řečeno, ukládání signálu je hlavní funkcí dataloggerů. Data poté mohou sloužit ke zpětné vizualizaci průběhu signálů, k jejich analyzování, vyhodnocení, zpracování. Pro spuštění zápisu do souboru je možné časované spuštění nebo trigger.

#### 3.3.3 Doplnkové funkce digitálního dataloggeru

Doplnkové funkce jsou jednou z výhod virtuálních dataloggerů oproti hardwarovým. Virtuální dataloggery tímto kopírují funkce dnešních digitálních osciloskopů s tou výhodou, že dané postupy, jako například filtrace, frekvenční analýza, matematické funkce, se provádějí až na načtený signál, což poskytuje výhodu oproti práci v reálném čase, kterou je neopomenutí důležitých momentů a nebýt omezován interní pamětí.

## 4 Programovatelné hradlové pole a Platforma CompactRIO

### 4.1 Programovatelné hradlové pole

#### 4.1.1 Základní vlastnosti Programovatelného hradlového pole

Programovatelná hradlová pole, všeobecně známá pod zkratkou “FPGA“, představují jeden z nejvýznamnějších směrů vývoje integrovaných obvodů s velmi velkou hustotou integrace (VLSI). Programovatelná hradlová pole jsou obvody s pravidelnou strukturou logických buněk schopných realizovat jednoduché logické funkce. Volitelným propojením těchto buněk lze dosáhnout rozsáhlých komplexních funkcí, k jejichž realizaci by jinak bylo nutné použít mnoho různých obvodů.

Programovatelná hradlová pole jsou velmi populární konstrukční alternativou k zákaznickým integrovaným obvodům, tzv. ASIC (Application Specific Integration Circuit). Důvodem jsou nízké výrobní náklady z pohledu koncového uživatele, velký počet vyrobených kusů z pohledu výrobce, krátká doba návrhového cyklu, možnost opakovaného použití (v případě obvodů FPGA využívajících technologii SRAM) a v poslední době též možnost dynamicky, za běhu, měnit konfiguraci celého FPGA nebo jeho částí. Konstrukce s FPGA naproti tomu dosahují menší výkonnosti (ve smyslu maximální pracovní taktovací frekvence či příkonu) než konstrukce s ASIC.

FPGA lze použít pro implementaci různých uživatelských zapojení (obvodů). Koncept univerzálního předpřipraveného obvodu, s nímž bude možné realizovat mnoho různých úloh, je velmi populární, neboť je to nejefektivnější způsob snižování výrobních nákladů. Existují dva typy takových obvodů: hradlová pole programovatelná maskou (Mask Programmable Gate Array – MPGA) a programovatelná hradlová pole (FPGA). Programovatelná hradlová pole byla odvozena přímo z MPGA. Prázdné MPGA je standardní čip, který obsahuje jednoduché logické prvky (tranzistory, hradla) a propojovací vodiče, určené především pro rozvod napájení umístěné v pravidelném vzoru na ploše čipu. Návrhář realizuje logický obvod v MPGA vytvořením masek kovových vrstev (metal layers) v posledním kroku výrobního procesu. To znamená, že koncový uživatel MPGA (návrhář) musí být součástí výrobního procesu. To na něj klade značné požadavky technické (schopnost dodržovat konstrukční pravidla a verifikovat navržené uspořádání) i ekonomické (dostatek finančních prostředků).

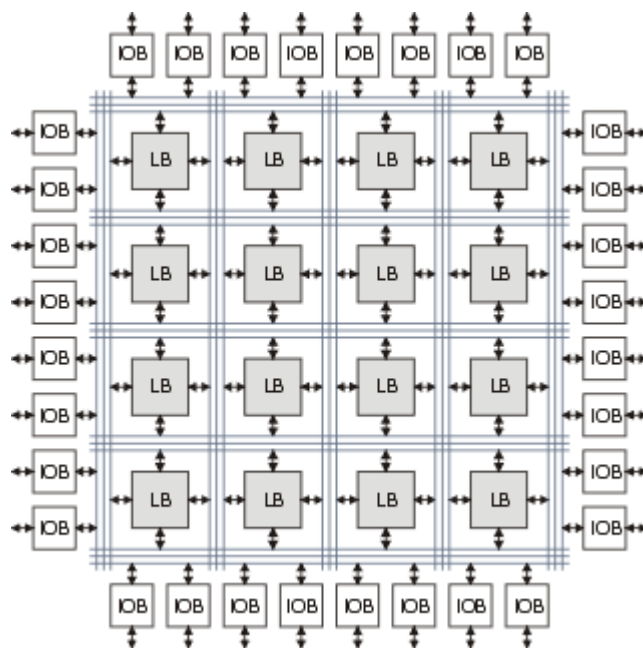
### 4.1.2 Architektura FPGA

Základní architektura obvodů FPGA obsahuje matici logických bloků (LB), velké množství lokálních a globálních propojovacích prostředků (programovatelné spojovací trasy PI), programovatelné vstupně/výstupní bloky (IOB), vstupně/výstupní oddělovače a statickou paměť konfigurace.

Vnitřní část obvodu FPGA implementuje v pravidelné architektuře LB bloky propojené rozsáhlou hierarchickou strukturou propojovacích vodičů. Vnější obvod programovatelné struktury tvoří vstupně/výstupní bloky.

Obvod se uvádí do činnosti nahráním konfiguračních dat do vnitřní statické paměti. Počet mazání a nahrávání dat není omezen. FPGA může číst konfigurační data aktivně z externí paměti – režim Master Seriál, nebo se zápis také provádí pomocí externího zařízení (např. procesor) – režim Slave Serial. Vlastní vkládání dat probíhá sériově nebo paralelně.

FPGA umožňuje vytvářet logické návrhy s hodnotou systémové frekvence přes 80 MHz a vnitřním taktováním přes 150 MHz.



Obr. 1: Typická struktura obvodu FPGA

### 4.1.3 Programování hradlového pole

Vývojová prostředí jsou v současnosti nezbytným nástrojem pro práci s programovatelnými obvody. Vstupní údaje (popis vyvíjené konstrukce) je nutné zapsat ve formě, kterou je systém schopen převést na model této konstrukce. Ten je pak možno zpracovávat simulátorem k ověření jeho správnosti, časových parametrů a podobně, a dále ho syntetizérem a implementačním programem implementovat do cílového programovatelného obvodu.

Zápis vstupních údajů bývá nejčastěji textový – pomocí jazyků HDL (Hardware Description Language) nebo grafický – editory schémat, stavových diagramů a podobně. Grafické systémy vstupu bývají obvykle nepřenositelné na jiný systém, než je ten, v němž byl popis vytvořen. Z tohoto hlediska jsou výhodnější systémy textového vstupu, které jsou ve výrazně větší míře standardizovány. K nejznámějším jazykům HDL patří jazyk ABEL a jazyk VHDL. Jazyk ABEL je poměrně jednoduchý, a jeho syntaxe vychází ze struktury obvodů PLD, pro něž je určen. Jazyk VHDL má výrazně vyšší stupeň abstrakce (a také složitosti), což dovoluje i syntézu zaměřenou na obvody FPGA.

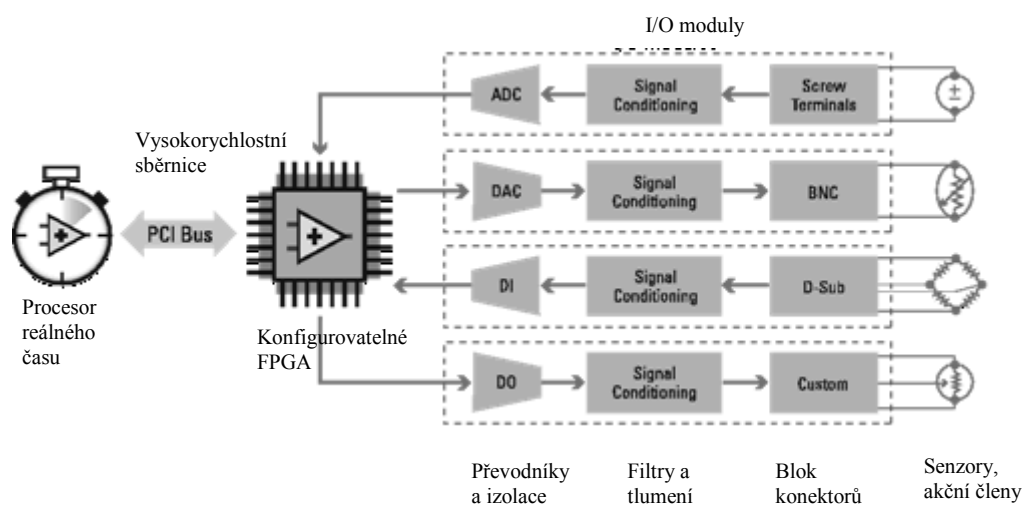
## 4.2 Platforma CompactRIO firmy National Instruments

CompactRIO je konfigurovatelný vestavěný systém pro robustní, nezávislé nebo síťové aplikace určené pro řízení a sběr dat. Systém se skládá z kontroléru reálného času, konfigurovatelného šasi, které obsahuje uživatelsky programovatelné hradlové pole FPGA, a z mnoha druhů průmyslových vstupně/výstupních modulů vyměnitelných za provozu.

Programovatelný automatizační kontrolér CompactRIO firmy National Instruments je moderní vestavěný systém pro řízení a sběr dat navržený pro aplikace požadující vysoký výkon a spolehlivost. K hlavním výhodám systému patří otevřená architektura, malé rozměry, extrémní odolnost a přizpůsobivost aj. NI cRIO funguje na technologiích National Instruments LabVIEW FPGA a LabVIEW Real Time, které dávají uživateli možnost navrhnout, vyvíjet a uvádět do provozu vestavěné systémy na míru pro širokou škálu odvětví (automobilový průmysl, vojenské a letecká technika, řízení průmyslových technologií a strojů, tvorba prototypů embedded zařízení a další) s využitím jednoduchých grafických programovacích nástrojů.

CompactRIO spojuje vestavěný procesor reálného času, výkonné FPGA a za provozu vyměnitelné I/O moduly. Real-Time Processor (procesor reálného času) využívá ke své funkci operační systém VxWorks. Každý modul je přímo připojen k FPGA za předpokladu nízké úrovně časování a I/O zpracování signálu na základě přizpůsobení systému potřebám uživatele. FPGA je připojeno k procesoru reálného času přes vysokorychlostní PCI sběrnici. Představuje tedy levnou architekturu s přímým přístupem k nízkoúrovňovému hardwarovému zdrojům. LabVIEW využívá vestavěné mechanismy k přenosu dat procházejících od I/O modulů přes FPGA až do procesoru, kde dochází k případné analýze, záznamu dat, dodatečnému zpracování nebo ke komunikaci s PC pomocí LAN sítě.





Obr. 2: Vnitřní struktura platformy CompactRIO

#### 4.2.1 Charakteristické rysy

- Malý, robustní, vestavěný systém určený pro aplikace vyžadující vysokou spolehlivost, malé rozměry, nízkou cenu a velký výkon.
- Využívá grafický programovací nástroj LabVIEW firmy National Instruments k rychlému vývoji aplikací.
- Charakteristickým rysem vestavěného procesoru reálného času jsou nezávislé nebo distribuované operace.
- Integrovaný FPGA čip poskytuje uživatelskému hardwaru flexibilitu, výkon a spolehlivost.
- Obsahuje průmyslové vstupně/výstupní (I/O) za provozu vyměnitelné moduly s vestavěným obvodem pro úpravu signálu, sloužící k přímému připojení různých senzorů a akčních členů.

#### 4.2.2 Parametry

- provozní teplota -40 až 70°C,
- odolnost až do 2300 V RMS izolovaného napětí,
- certifikace ukazující nezávadnost k životnímu prostředí, elektromagnetická kompatibilita,
- použití i v prostředí s možností výbuchu,
- duální napájecí vstup 9 až 35V DC,
- odolnost vůči nárazu 50g.

### 4.2.3 Možnosti použití

- řízení, získávání a záznam dat v automobilovém průmyslu,
- monitorování a ochrana stavu strojů,
- vzdálené a distribuované monitorování,
- vestavěné prototypy, záznam dat,
- uživatelské řízení pohybu,
- mobilní/kufříkový analyzátor hluku, vibrací, drsnosti,
- řízení v těžkém průmyslu,
- dávkové a diskrétní řízení,
- sledování elektrického napájení a výkonové ovládání.

### 4.2.4 Kontrolér

Programovatelný automatizační kontrolér CompactRIO nabízí široké možnosti v oblasti řízení a sběru dat v kompaktním a robustním odolném provedení. Obsahuje malý, robustní, vysoce spolehlivý vestavěný procesor (Real-Time Processor) určený pro spolehlivé vykonávání samostatných aplikací. Vestavěný řadič, běžící v programovacím prostředí LabVIEW, slouží pro deterministické řízení, záznam (sběr) dat, jejich analýzu a komunikaci.

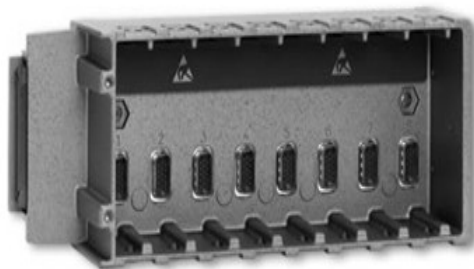


Obr. 3: Kontrolér reálného času NI 9104

### 4.2.5 FPGA (šasi)

CompactRIO šasi je srdcem celého systému, protože spojuje konfigurovatelné vstupně/výstupní bloky (RIO) jádra FPGA. Jádro má připojení každého I/O modulu individuální a je programováno s využitím jednoduchých základních I/O funkcí ke čtení nebo zápisu signálu každé karty. Protože komunikační sběrnice mezi FPGA jádrem a I/O moduly není sdílená, musí být I/O operace každé karty naprosto přesně synchronizovány s 25ns rozlišením. Přenos informací je mezi RIO jádrem a kontrolérem zajištěn pomocí PCI rozhraní.

FPGA je velmi výkonný čip konfigurovatelný pomocí nástrojů LabVIEW FPGA. V minulosti bylo při vytváření aplikace na FPGA nutné bezpečně znát jazyk VHDL. Nyní tuhle práci zajišťuje v LabVIEW integrovaný překladač. Použití FPGA umožňuje realizovat uživatelské časování, triggrování, synchronizaci, řízení a zpracování signálu pro analogové a digitální I/O karty.



Obr. 4: Konfigurovatelné FPGA šasi NI 9104

#### 4.2.6 I/O moduly

I/O moduly slouží k připojení velkého množství typů signálů, ke kterým patří např.: napěťové, proudové vstupy, termočlánky, akcelerometry a tenzometry, simultánní vzorkovací analogové I/O až do  $\pm 60V$ , 12, 24 a 48V průmyslové digitální I/O, 5V/TTL digitální I/O, čítače/časovače, pulzní generátory, vysoko napěťové a proudové přenosy. Protože moduly obsahují vestavěný obvod pro úpravu signálů k rozšíření rozsahu napětí nebo pro průmyslové typy signálů, je tedy možné pomocí kabelů přímo propojit moduly se senzory a akčními členy. I/O moduly je možné měnit za provozu.



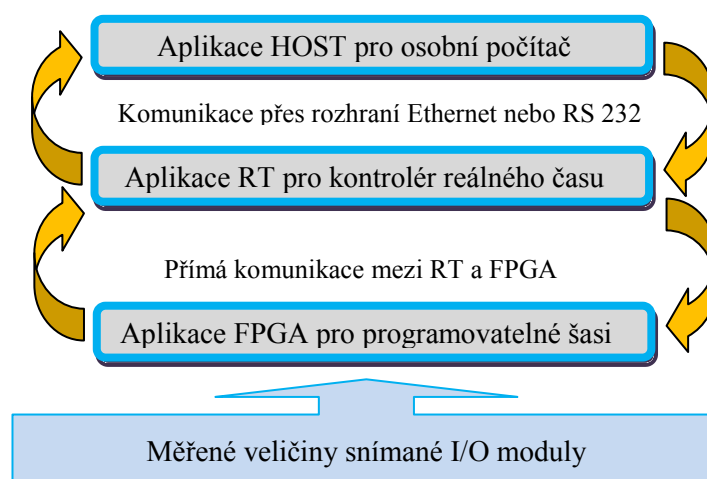
Obr. 5: Vstupní/Výstupní moduly

### 4.2.7 Vývoj aplikací pro platformu CompactRIO

Návrh softwaru pro platformu CompactRIO se mírně liší od návrhu virtuálních přístrojů na multifunkčních kartách nebo na systému CompactDAQ. Je to vnitřní strukturou platformy CompactRIO (viz obrázek 3).

Hardwarová architektura umožňuje přístup k měřenému signálu pouze přes hradlové pole. Poněvadž CompactRIO neobsahuje hardwarově řešené uživatelské rozhraní (klávesnice, obrazovka), je nutné pro komunikaci s kontrolérem, resp. s FPGA využít tzv. Host PC, na kterém je možno uživatelské rozhraní spustit.

Aplikace pro CompactRIO, je složena z kódu pro FPGA, na kterém běží časově kritické smyčky, kódu pro kontrolér a virtuálního přístroje pro grafické uživatelské rozhraní. Komunikace mezi GUI a kontrolérem probíhá pomocí síťového propojení (např. TCP/IP). Platforma také umožňuje přímou komunikaci mezi GUI (HostPC) a kódem běžícím na FPGA.



Obr. 6: Architektura aplikací na platformě CompactRIO

#### 4.2.7.1 Aplikace Host pro osobní počítač

První vrstva aplikační struktury pro cRIO se nachází na osobním počítači a obsahuje grafické komunikační rozhraní, které slouží pro ovládání zařízení uživatelem a prezentací dat. Dále tyto aplikace obsahují výpočetně náročné funkce, např. zpracování údajů poskytnutých z nižších vrstev aplikace. Komunikace s CompactRIO probíhá přes rozhraní Ethernet. Programování aplikace pro PC je možné využít pro všechny funkce prostředí NI LabVIEW.

#### 4.2.7.2 Aplikace RT pro kontrolér reálného času

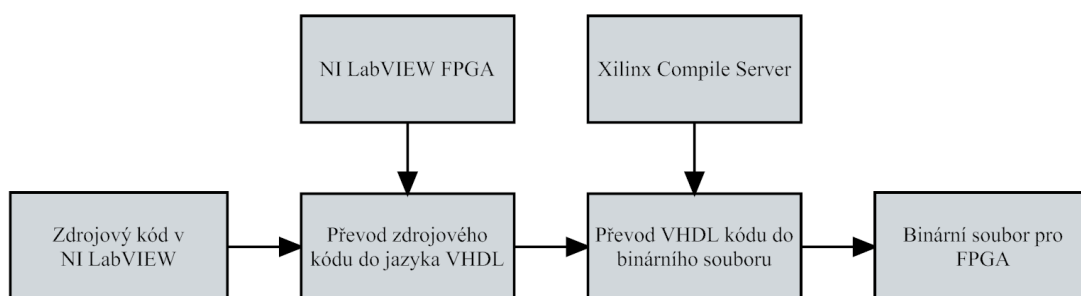
Prostřední vrstva struktury může sloužit jako prostředník mezi HOST a FPGA, nebo jako samostatný vestavěný systém umožňující samostatné fungování systému. Kontrolér získává data z ovládacích, indikačních prvků nebo z DMA FIFO paměti. Zdrojový kód určený

pro kontrolér se vyvíjí v modulu nazvaném NI LabVIEW RT. Důvodem je použitý RTOS Wind River VxWorks. Nabídka typů proměnných a knihovních funkcí je srovnatelná s PC platformou. Je nutné respektovat, že RT kontrolér nemá takový výkon jako běžný PC, proto je nutné optimalizovat kód nebo data pouze předzpracovávat a složitější výpočty vykonávat až v PC aplikaci. Aplikace komunikuje přes sběrnici s FPGA aplikací a přes rozhraní Ethernet nebo RS232 s aplikací HOST.

#### 4.2.7.3 Aplikace FPGA na programovatelné šasi

Na úrovni FPGA je možné využít funkce pro základní práci se vstupy a výstupy. Pro možnost programování FPGA čipů v NI LabVIEW je nutné mít nainstalován NI FPGA modul, který obsahuje kompilátor VHDL kódu do binárního souboru, který se nahraje do FPGA čipu. Při programování FPGA v NI LabVIEW je nutné vědět následující:

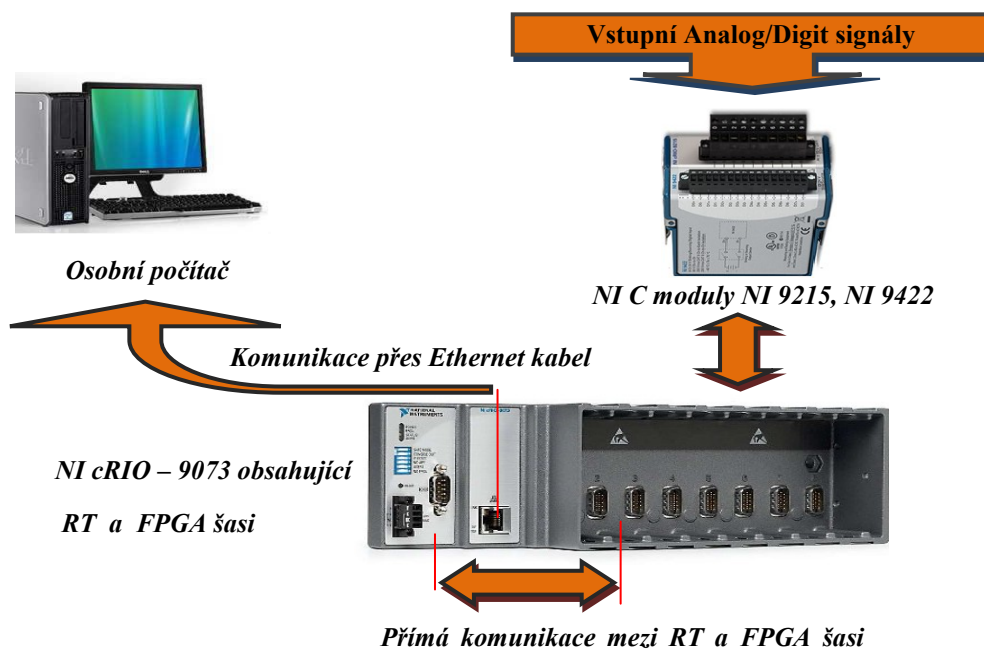
- Používat pouze celočíselné datové typy (FPGA nepodporuje datové typy s pohyblivou desetinnou čárkou).
- Je možné použít pouze omezené množství knihovních funkcí.
- Pro časování smyček jsou dostupné pouze určité specifické frekvence odvozené od základních hardwarových hodin (40MHz).
- Nemožnost použití standardních prostředků pro ladění kódu.
- Pro transport dat mezi FPGA a kontrolérem je možné používat paměť FIFO (počet FIFO pamětí je omezen) nebo přímé snímání.
- Omezené množství hardwarového místa, je nutná optimalizace zdrojového kódu.
- Možnost využití integrovaných hardwarových bloků v FPGA čipu jako jsou např. násobičky a paměti.



Obr. 7: Převod zdrojového kódu do binárního souboru pro FPGA

## 5 Realizace hardwarové části virtuálního dataloggeru

Virtuální datalogger je založen na bázi National Instruments CompactRIO. Využívá osobního počítače, NI CompactRIO integrovaný systém kombinující procesor reálného času a konfigurovatelné programovatelné hradlové pole, vstupně výstupní NI C moduly. Osobní počítač obsahuje grafické uživatelské rozhraní, které slouží uživateli jako prezentační a ovládací zařízení celého systému. PC komunikuje s CompactRIO RT přes 100Mb Ethernet pomocí TCP/IP protokolu. CompactRIO RT slouží jako prostředník mezi PC a FPGA šasi. RT se dá rozdělit na dvě části; v první přijímá rozkazy a posílá snímaná data přes Ethernet, v druhé předává rozkazy FPGA šasi a odebírá data z DMA FIFO paměti. CompactRIO RT a FPGA jsou propojeny přímo. FPGA snímá jednotlivé vzorky měřeného signálu do paměti DMA FIFO ze všech snímaných vstupů zároveň.



Obr. 8: Schéma zapojení virtuálního dataloggeru

## 5.1 Technické parametry využitých hardwarových komponentů

### 5.1.1 Parametry kontroléru NI cRIO – 9073

National instruments cRIO-9073 integrovaný systém kombinuje procesor reálného času a konfigurovatelné programovatelné hradlové pole (FPGA) v jednom zařízení pro vestavěné kontrolní a monitorující aplikace. Integruje 266 MHz průmyslový procesor reálného času s dvěma miliony hradel FPGA a má 8 slotů pro NI vstupně/výstupní moduly série C. Pro robustní aplikace nabízí operační teploty od -20 do 55 °C se stejnosměrným vstupním napětím 19 - 30 voltů. Nabízí 64 MB DRAM systémové paměti a 128 MB uživatelské paměti přístupné aplikacím pro ukládání dat. S 10/100MB Ethernet portem lze řídit programovatelnou komunikaci přes síť s vestavěným webovým (HTTP) a souborovým serverem.



Obr. 11: Kontrolér NI cRIO – 9073

#### Technické parametry NI cRIO-9073:

- 8 slotů pro připojení vstupně/výstupních modulů,
- 266 MHz průmyslový procesor,
- 64 MB DRAM systémové paměti,
- 128 MB uživatelské paměti, energeticky nezávislé,
- 2 miliony hradel FPGA,
- 40MHz generátor hodinového signálu, možnost skokového snížení nebo zvýšení frekvence,
- 10/100 MB Ethernet,
- RS-232 sériový port 115, 200 bps,
- 19-30V vstupní napětí, maximální spotřeba 20W,
- -20 do 55 °C operační teplota.

## 5.1.2 Parametry C modulů NI 9215, NI 9422

### 5.1.2.1 Modul analogových vstupů NI 9215

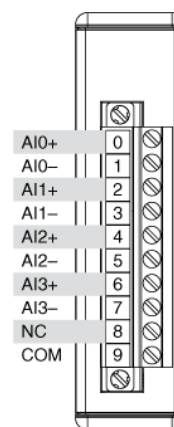
Analogový vstupní modul s paralelním vzorkováním použitelný v NI CompactDAQ a CompactRIO šasi obsahuje 4 paralelně vzorkované vstupní kanály s 16 bitovým vzorkováním. NI 9215 obsahuje NIST – kalibrace, dvojitou kanál-zem izolaci pro bezpečnou a šumovou imunitu.

#### Technické parametry NI 9215:

- 4 současně vzorkované analogové vstupy  $\pm 10\text{V}$ ,
- 16 bitový aproximační A/D převodník,
- maximální vzorkovací frekvence 100 000 vzorků/s na všech kanálech,
- možnost vysunutí / zasunutí modulu za běhu,
- rozsah pracovních teplot  $-40$  až  $70^\circ\text{C}$ ,
- izolované vstupy,
- maximální napětí  $\pm 30\text{V}$ ,
- střední doba mezi poruchami 1 167 174 hodin,
- maximální spotřeba 560 mW.



Obr. 9: Analogový vstupní C modul NI 9215



Obr 10: Zapojení vstupů NI9215

### 5.1.2.2 Modul digitálních vstupů NI 9422

NI 9422 je osmi kanálový digitální vstupní modul. Každému kanálu vyhovuje od 24 až do 60V digitální vstupní signál a nabízí kanál-kanál izolaci pro nejvyšší úroveň bezpečnosti v automobilovém a průmyslovém prostředí. Každá diskrétní úroveň signálu



připojená k NI 9422 pod 5V je registrována jako digitální low a každý diskretní signál mezi 11 do 60V je registrován jako digitální High. Modul NI 9422 je korelovaný digitální modul, takže dokáže korelační operace, spouštění (trigger) a synchronizaci, pokud je instalovaný na NI CompactDAQ šasi.

#### Technické parametry NI 9215:

- 8 digitálních vstupů,
- 11-60V digitální logika,
- rychlost snímání vstupu,  
z 0 na 1 250μs max, 4μs typické  
z 1 na 0 250μ max. 130μs typické,
- možnost vysunutí / zasunutí modulu za běhu,
- rozsah pracovních teplot -40 až 70°C,
- střední doba mezi poruchami 1 220 439 hodin,
- maximální spotřeba 55 mW.



Obr. 11: Digitální vstupní modul NI 9422

Module	Terminal	Signal
	0	DI 0+
	1	DI 0-
	2	DI 1+
	3	DI 1-
	4	DI 2+
	5	DI 2-
	6	DI 3+
	7	DI 3-
	8	DI 4+
	9	DI 4-
	10	DI 5+
	11	DI 5-
	12	DI 6+
	13	DI 6-
	14	DI 7+
	15	DI 7-

Obr. 12: Zapojení vstupů NI 9422

### 5.1.3 Osobní počítač Lenovo T 61

I když lze navržený virtuální datalogger využívat na všech osobních počítačích alespoň minimální uvedené konfigurace, uvádím zde i vlastní počítač využitý pro vývoj a ověření funkcí dataloggeru. Jedná se o notebook Lenovo T 61.

#### Technické parametry Lenovo T 61:

- procesor Intel dual core T 8300 2,4 Ghz,
- 4 GB operační paměti,
- 500 GB, 7200 otáček, pevný disk,
- nVidia Quadro 140m 128MB,
- 15.4" WSXGA+ 1680x1050 matný displej,
- optická mechanika: DVD-ROM 8x MaxUltrabay Slim,
- síť: Intel 802.11abg, Integrated Intel Gigabit Network Adapter, 56Kbps data,
- porty: 3x USB 2.0, RJ-45, RJ-11, IEEE 1394, sluchátka / Line out, externí mikrofon Line-In, externí displej (VGA), PCMCIA slot a Express Card slot.



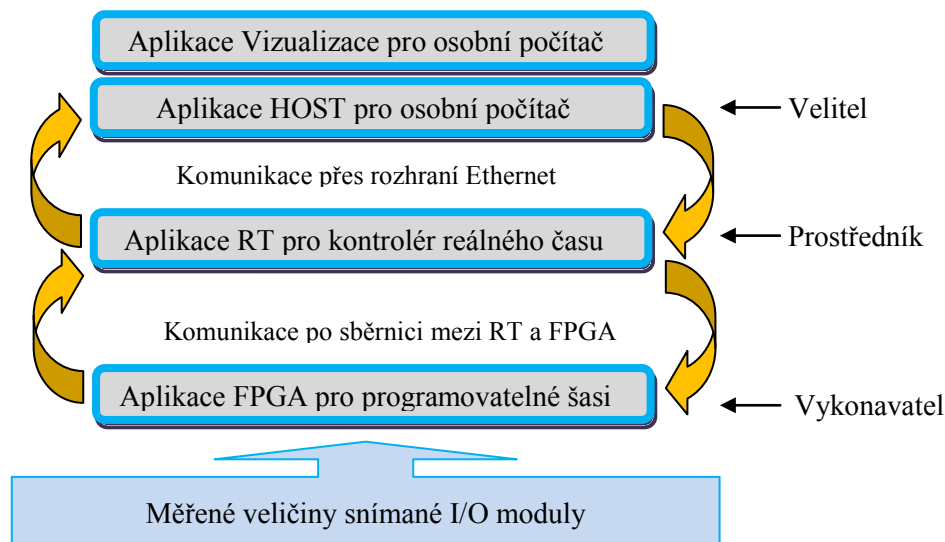
Obr. 13: Notebook Lenovo T61

## 6 Realizace softwarové části virtuálního dataloggeru

Virtuální datalogger realizovaný v této práci je konstruován na bázi virtuální instrumentace ve vývojovém prostředí National Instruments LabView 2009 s využití modulů:

- NI FPGA 9.0,
- NI Real-Time 9.0,
- Labview Run-Time 2009,
- NI RIO 3.2.0,
- NI MAX 4.6,
- a dále knihoven SMT (jednoduché přenášení zpráv) a TDMS (Správa technických dat).

Aplikace jsem pojmenoval podle jejich umístění, v jednom případě podle účelu. Datalogger je konstruován podle vývojové architektury platformy CompactRIO, jednotlivé části virtuálního dataloggeru je nutné realizovat jako samostatné aplikace. Tyto aplikace mezi sebou komunikují na základě příkazů. Vzhledem k tomu, že velké množství dat najednou je schopno člověka zahltit, bude aplikaci vhodné rozdělit do několika vnořených aplikací, vykonávající dílčí úlohy dataloggeru.



Obr. 14: Hierarchie aplikace Virtuální datalogger

### 6.1.1 Komunikace na základě příkazu

Komunikace na základě příkazů je něco, co se děje relativně zřídka, a je spouštěna na základě konkrétní události. Příkladem je uživatel, když stiskne tlačítko na ovládacím rozhraní zařízení ke spuštění měření. Zdrojem příkazů je velitel, v mém případě aplikace „HOST“, která zasílá příkazy prostředníkovi, aplikaci „RT“. Ta předá příkaz vykonavateli aplikaci „FPGA“.

### 6.1.2 Návrh aplikace z pohledu uživatele

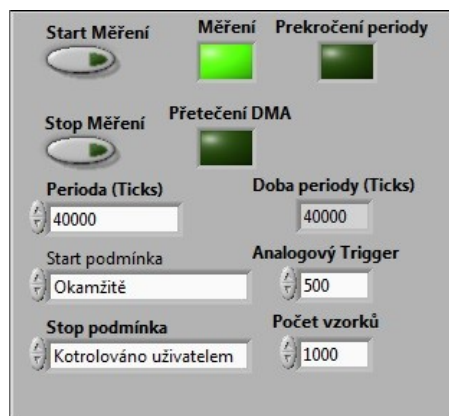
Aplikace by měla být přehledná, srozumitelná a s jednoduchým intuitivním ovládáním.

Z pohledu uživatele lze aplikaci rozdělit na tři základní části.

- konfigurace,
- zaznamenávání, zobrazení a zpracování průběhu signálů v reálném čase,
- zpětná vizualizace průběhu signálů uložených na pevném disku.

## 6.2 Aplikace FPGA

Aplikace FPGA slouží ke snímání měřených veličin připojených ke vstupům C modulů. Aplikace je plně řízena, spouštěna i ukončována aplikací RT. Uživatel nemá k této aplikaci za normálních okolností přístup.



Obr. 15: Přední panel aplikace FPGA

Aplikace využívá k ukládání snímaného signálu dvě DMA FIFO paměti, každá je nastavena pro uložení 16 383 proměnných. První pro analogové signály datového typu Unsigned 32(32 bitového nezáporného celého čísla), druhá pro digitální signály datového typu Boolean (0,1).

Po spuštění aplikace provede aplikace RT nastavení aplikace, následuje čekání na spuštění měření. Spuštění aplikace se odvíjí od nastavení startovních podmínek. Po spuštění měření jsou sejmuty jednotlivé vzorky měřených signálů ze vstupů C modulů v závislosti na vzorkovací frekvenci. Snímání analogových i digitálních signálů probíhá současně. Po sejmutí jsou tyto vzorky uloženy do FIFO paměti. Ukončení aplikace opět závisí na nastavených podmínkách. V případě, že RT aplikace nedokáže v dostatečné rychlosti odebírat uložené vzorky a dojde-li k přetečení FIFO paměti, aplikace se automaticky vypne.

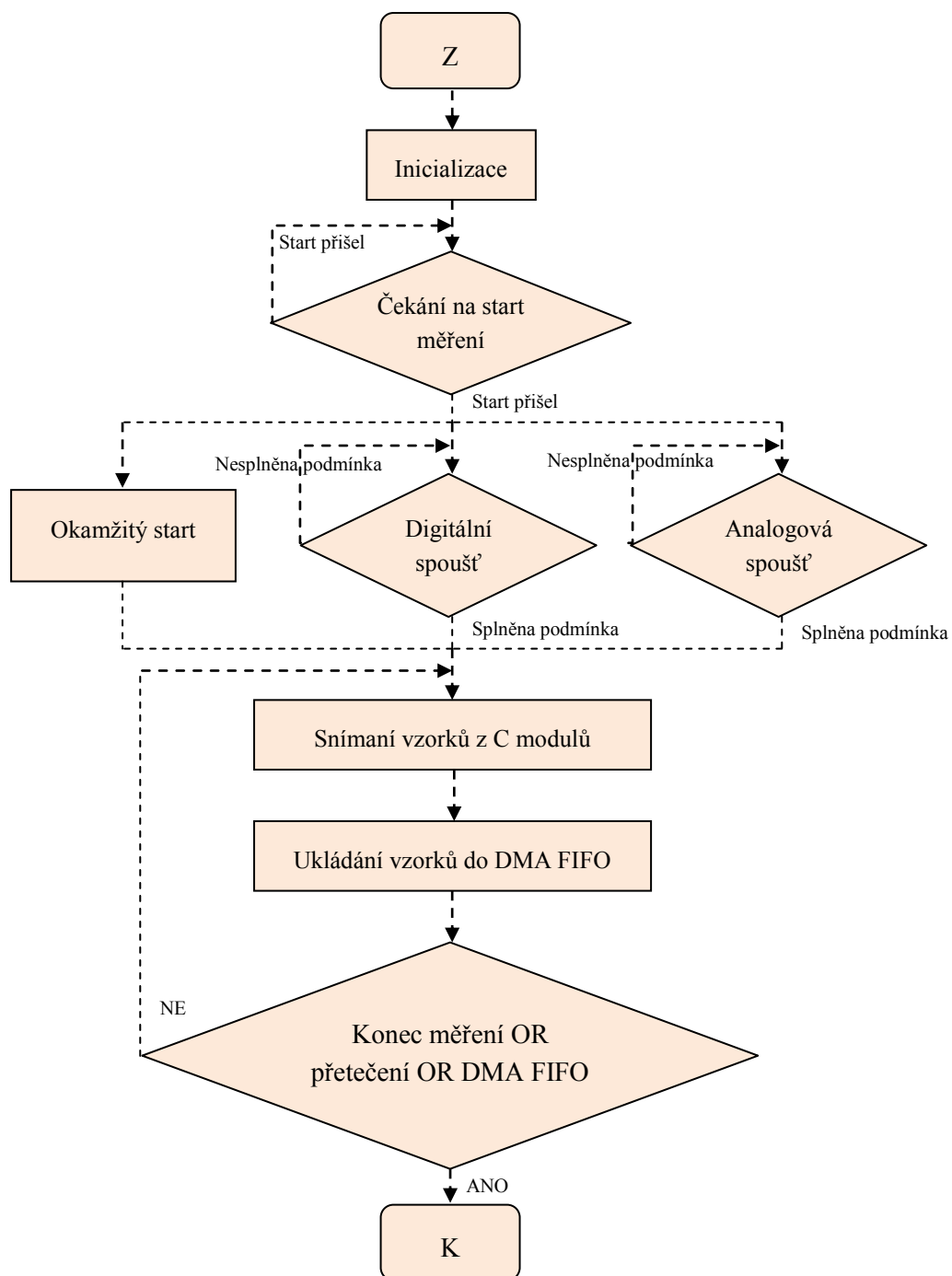
K realizaci vzorkovací frekvence je využit 40MHz generátor hodinového signálu. Aplikace v průběhu činnosti čítá jednotlivé periody generátoru a porovnává se zadanou periodou. Ta je vypočtena podle Rovnice 2: Celkový počet odebíraných vzorků. Tuto hodnotu vypočítává aplikace RT.

$$t_s = \frac{40000000}{f_s} [\text{Period generátorů hodinového signálu}]$$

Rovnice 1: Výpočet vzorkovací periody

kde  $t_s$  perioda vzorkování  
 $f_s$  frekvence vzorkování

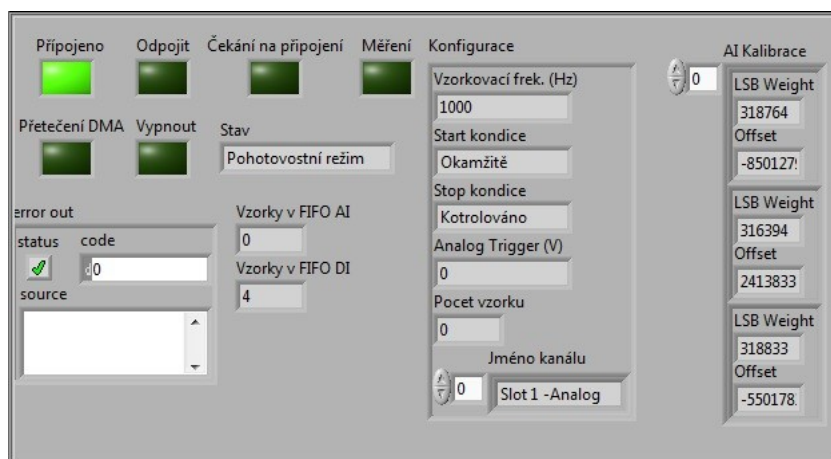
40 000 000 frekvence generátoru hodinového signálu



Obr. 16: Vývojový diagram FPGA aplikace

### 6.3 Aplikace RT

Aplikace slouží jako prostředník mezi aplikacemi FPGA a HOST. Aplikace je konstruována ke zjištění vysoké rychlosti průběhu, proto obsahuje jen ty nejdůležitější indikační prvky. Po svém startu aplikace čeká na přepnutí USER 1 přepínače do stavu zapnuto. Po přepnutí dojde ke spuštění vnořené aplikace “Načtení kalibrace” pro získání kalibračních hodnot jednotlivých vstupů modulů NI 9215. Zároveň se čeká na připojení od aplikace HOST přes Ethernet. Tento stav čekání na připojení lze zjistit z indikační LED diody USER 1 nacházející se na přední straně cRIO. Po úspěšném připojení aplikace dojde k odeslání Meta dat potřebných ke komunikaci přes STM komunikační protokol, poté dojde k odeslání kalibračních hodnot. V další fázi se program dělí na dva oddělené cykly – první pro komunikaci s aplikací FPGA, druhý pro komunikaci s aplikací HOST. Příkazy přijaté od aplikace HOST jsou vkládány do RT FIFO paměti, odkud jsou následně odebrány a plněny smyčkou FPGA. Nevýhodou RT FIFO paměti je, že dokáže uchovat pouze číselné datové typy (Integer, Double, Atd..). Proto se měřená data mezi smyčkami předávají pomocí funkce Fronty, obdoba FIFO paměti bez omezení datového typu. Fronta je tvořena datovou strukturou Cluster, která se skládá z pole datových typů Integer16 dále pole datových typů Boolean a datové struktury Times stamp ( Časová známka). Fronta uchovává 10 datových struktur. Uživatel nemá k této aplikaci za normálních okolností přístup.



Obr. 17: Přední panel Aplikace RT

### 6.3.1 Cyklus komunikace s FPGA

Prioritní, časově kritický cyklus s účelem kontroly a sběru dat z aplikace FPGA. Perioda opakování cyklu je 10 ms, z čehož vychází obnovovací frekvence celého systému. Cyklus odebírá příkazy z FIFO paměti, vkládané HOST smyčkou. Cyklus se dále dělí na jednotlivé stavy, které jsou volány v závislosti na příkazech.

#### 6.3.1.1 Stav Pohotovostní stav

Jedná se o nekonečný cyklus čekající na příkaz. Po přijmutí příkazu přepíná do dalších stavů Konfigurace, Start, Shutdown.

#### 6.3.1.2 Stav Inicializace

V prvotní fázi spouští aplikaci FPGA. Poté přepnutí do „Pohotovostním stavu“.

#### 6.3.1.3 Stav Konfigurace

Nastavení aplikace FPGA podle přijaté konfigurace. Přepočet hodnoty Analogového spouště vnořenou aplikací Kalibrace, vzorkovací frekvence na vzorkovací periodu podle Rovnice 1: Výpočet vzorkovací periody. Výpočet počtu odebraných vzorků za obnovovací periodu podle Rovnice 2: Celkový počet odebíraných vzorků, počet vzorků jednoho signálu podle Rovnice 3: Počet vzorků jednoho signálu. Poté přepnutí do „Pohotovostním stavu“.

$$\text{Počet odebíraných vzorků} = \frac{\text{Vzorkovací frekvence}}{\text{Frekvence obnovení systému}} * \text{počet signálů [Vzorků]}$$

*Rovnice 2: Celkový počet odebíraných vzorků*

$$\text{Počet odebíraných vzorků} = \frac{\text{Vzorkovací frekvence}}{\text{Frekvence obnovení systému}} [\text{Vzorků}]$$

*Rovnice 3: Počet vzorků jednoho signálu*

#### 6.3.1.4 Stav Start

Inicializace DMA FIFO paměti. Započetí měření udělením příkazu „Start Měření“ aplikaci FPGA. Poté přepnutí do stavu „Běh“.

#### 6.3.1.5 Stav Běh

Nejdůležitější funkce této smyčky. Slouží pro odběr vzorků signálů z DMA FIFO paměti „AI Data“ a „DI Data“. Problémem je, že FPGA neumí pracovat s polem hodnot, respektive do paměti FIFO je možné zapisovat pouze po jedné hodnotě nezáporných celočíselných datových typů. Při vyčtení vzorků z FIFO paměti jsou naměřené hodnoty ve formě pole datových typů, pro které dále pomocí funkce Přebudování polí se transformuje do 2D datového pole o 4 dat.polích jedno datové pole pro jeden signál. V případě analogového signálu je změněn datový typ z 32 Unsigned na 16 Integer vzhledem k rozlišení 16 bitového



A/D převodníku modulu NI 9215. Z 2D datového pole jsou vybrány pouze datová pole zvolených signálů, ostatní jsou zahozena. V další fázi je vytvořena časová známka, která je společně s analogovými i digitálními vzorky vložena do Fronty v podobě datové struktury klastr. Cyklus běží do přijetí příkazu „Stop“ nebo chybě v pamětech DMA FIFO. Chyba žádného nebo nedostatku uložených vzorků v paměti je ignorována. Poté přepnutí do stavu „Stop“

#### 6.3.1.6 Stav Stop

Vydá povel aplikaci FPGA stop měření. Poté přepnutí do „Pohotovostním stavu“.

#### 6.3.1.7 Stav Shutdown

Příkaz vymaže používanou FIFO paměť a Frontu, vypne aplikaci FPGA i celý cyklus komunikace s FPGA.

### 6.3.2 Cykly komunikace s Host

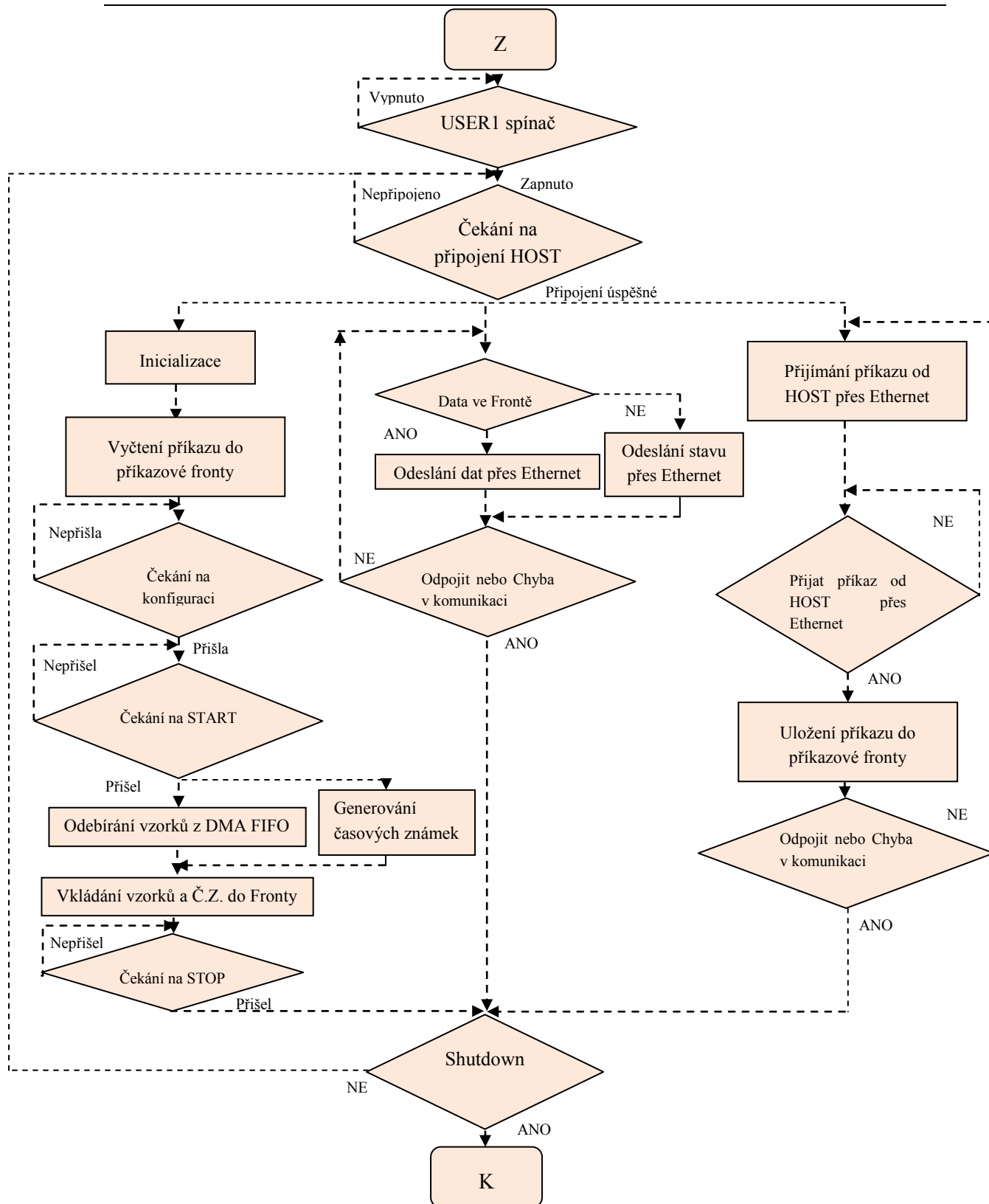
Prioritně nižší cykly komunikující s aplikací Host přes síťové rozhraní Ethernet pomocí funkcí STM.

#### 6.3.2.1 Cyklus přijímání dat přes Ethernet

Prioritně nejnižší cyklus přijímá příkazy a konfigurační klastr. Cyklus se opakuje po 100 ms s ohledem na výpočetní výkon. Určení jednotlivého typu zpráv se provádí podle jména Meta prvku. Jednotlivé příkazy se vkládají do RT FIFO. Konfigurační klastr se uloží do indikačního prvku „Konfigurace“ odkud je volán přes lokální proměnnou. Ukončení cyklu po příkazu „Odpojit“ nebo chybě ve spojení.

#### 6.3.2.2 Cyklus odesílání dat přes Ethernet

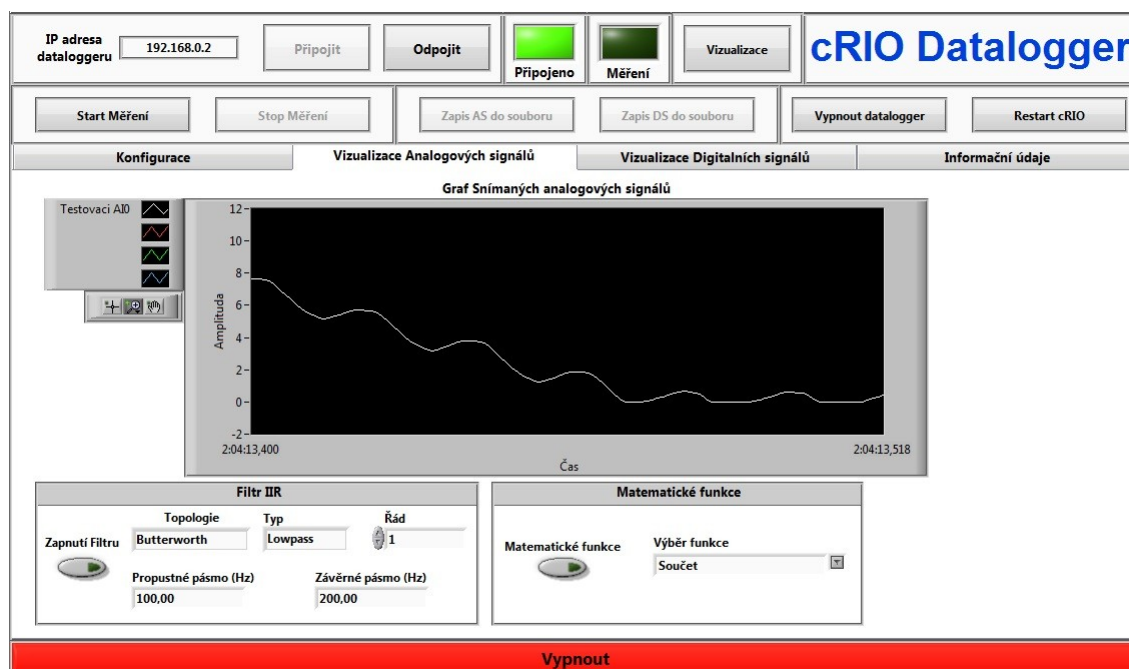
Cyklus kontroluje počet prvků ve Frontě. Při jednom a více prvcích dojde k odeslání těchto prvků přes síťové rozhraní s Meta prvkem „Data“. Při žádném prvcích ve Frontě dojde k odeslání hodnot některých důležitých indikačních prvků Meta prvkem „Stav“. Cyklus se opakuje každou 1 ms, rychleji než jejich vkládání do Fronty a zajištění přenosu stavu i při velkém vytížení.



Obr. 18: Vývojový diagram RT aplikace

## 6.4 Aplikace HOST

Aplikace HOST slouží uživateli jako ovládací, indikační a prezentační část vytvořeného projektu na platformě CompactRIO. Uživatele nutí k dodržení postupu měření tím, že nedovoluje přístup k ovládacím prvkům, pokud nejsou splněny podmínky pro jejich použití (např. start měření bez nastavení konfigurace). Přední panel aplikace se dělí na panel tabelových oken a horní panel.



Obr. 19: Přední panel aplikace HOST

### 6.4.1 Prvky horního panelu

Obsahuje ovládací prvky, které jsou nezávislé na zvoleném tabelovém okně, jako komunikace s cRIO, zapnutí/vypnutí měření a zápisu do souboru, vypnutí dataloggeru, restartování CompactRIO. Přístupnost k některým prvkům je omezena v závislosti na jejich využití.

- **IP adresa dataloggeru:** nastavení adresy síťového rozhraní cRIO,
- **Připojit / Odpojit:** připojení/odpojení aplikace ke cRIO,
- **Připojení:** indikuje stav spojení,
- **Měření:** indikuje stav měření,
- **Vizualizace:** spouští aplikaci „Vizualizace“.

- **Start / Stop měření:** zapíná/vypíná měření. Prvky zpřístupní pouze po správném zadání konfigurace.
- **Zápis do AI/DI souboru:** zapíná/vypíná zápis AI/DI do souboru. Prvky zpřístupní pouze po správném zadání cesty k uložení.
- **Vypnout datalogger:** vypne aplikaci RT na cRIO, které je poté nutné restartovat pro opětovné spuštění aplikace přístupné pouze po připojení k cRIO.
- **Restartování cRIO:** restartuje cRIO na adrese zadání v prvku **IP adresa dataloggeru**

### 6.4.2 Prvky tabelového okna

Tabelové okno obsahuje záložky pro volbu jednotlivých oken obsahujících jednotlivé kontrolní a prezentační prvky. Toto opatření je zvoleno z velkého počtu prvků předního panelu aplikace.

#### 6.4.2.1 Okno Konfigurace

Nastavení

- **Výběr kanálů pro snímání:**
  - Zvolení kanálu pro snímání,
  - Indikace umístění kanálu *modul – označení kanálu*,
  - Nastavení jména kanálu.
- **Nahrát konfiguraci na cRIO:** odešle konfiguraci aplikaci RT přes Ethernet rozhraní
- **Vzorkovací frekvence:** nastavení vzorkovací frekvence snímání v jednotce Hertz
- **Start podmínka:** nastavení podmínky spuštění měření,
  - **Okamžitě** – spustí měření okamžitě po stisknutí startu měření.
  - **Digital trigger** – zahájení měření v závislosti na vzestupnou hranu signálu na vstupu digitálního modulu „DI5“.
  - **Analog. trigger** – po startu měření čeká na úroveň analogového signálu na vstupu AI0 a pokud je větší nebo rovna hodnotě zadané na následujícím prvku, měření je spuštěno.
- **Úroveň analog. triggeru (V):** nastavení hodnoty pro spuštění měření analogovou spouští. V jednotce Volt.
- **Stop podmínka:** nastavení podmínky pro vypnutí měření
  - **Kontrolování uživatelem** – vypnutí měření po stisknutí tlačítka „Stop Měření“

- **Digital trigger** – vypnutí měření v závislosti na sestupnou hranu signálu na vstupu digitálního modulu „DI5“.
- **Počet vzorků** – vypne měření po sejmutí počtu vzorků zadaného v následujícím prvku.
- **Zadat cestu k uložení AS:** otevře dialogové okno zadání cesty k uložení analogových signálů do souboru s příponou \*.tdms.
- **Cesta k souboru AS data:** zobrazí zvolenou cestu k uložení analogových signálů.
- **Zadat cestu k uložení DS:** otevře dialogové okno zadání cesty k uložení digitálních signálů do souboru \*.tdms.
- **Cesta k souboru DS data:** zobrazí zvolenou cestu k uložení digitálních signálů.
- **Uložit konfiguraci:** Uloží nastavenou konfiguraci do souboru s příponou \*.cfg.
- **Načíst konfiguraci:** Načte uloženou konfiguraci ze souboru s příponou \*.cfg.

#### 6.4.2.2 Okno Vizualizace Analogových signálů

- **Graf snímaných analogových signálů:** Prezentační prvek zobrazující průběh snímaných analogových signálů. Více o jeho ovládání v kapitole 6.6 Manipulace s grafy
- Filtrace**
- **Zapnutí filtru:** zapne funkci filtrace signálu,
  - **Topologie:** zvolení topologie filtru,
  - **Typ:** zvolení typu filtru,
  - **Řád:** nastavení řádu filtru,
  - **Propustné Pásmo (Hz):** nastavení propustného pásma filtru,
  - **Závěrné pásmo (Hz):** nastavení závěrného pásma filtru.

#### Matematické funkce

- **Matematické funkce:** zapnutí matematických funkcí,
- **Výběr funkce:** zvolení matematické funkce.

#### 6.4.2.3 Okno Digitálních signálů

- **Graf snímaných digitálních signálů:** Prezentační prvek zobrazující průběh snímaných digitálních signálů. Více o jeho ovládání v kapitole 6.6 Manipulace s grafy.

#### 6.4.2.4 Okno Informační údaje

V tomto tabelovém oknu jsou zobrazeny některé indikační prvky sloužící pro možné zjištění problémových stavů programu, informace o komunikaci.

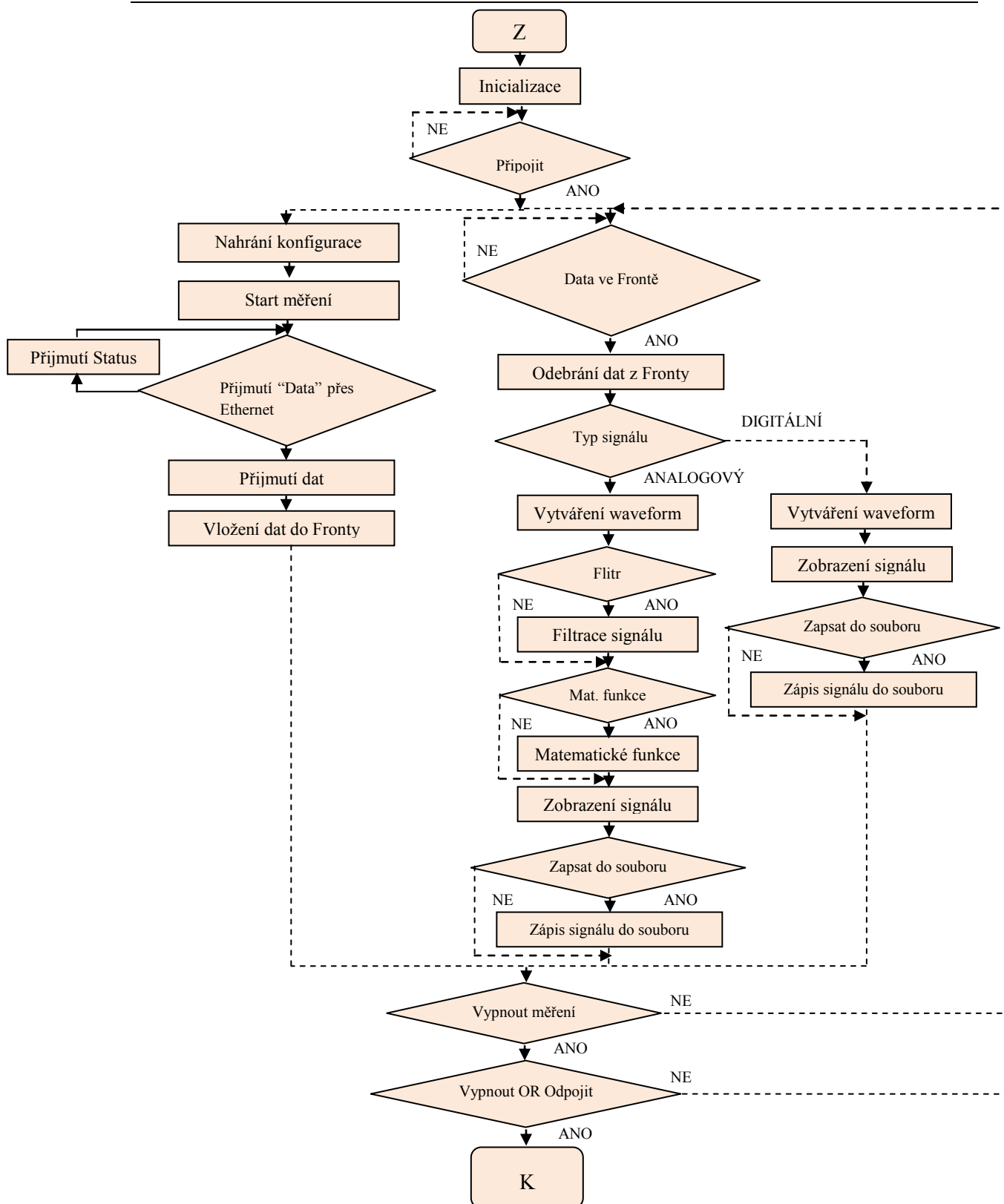
#### 6.4.3 Programová struktura aplikace HOST

Programová struktura se skládá ze tří časovaných cyklů závislých na připojení aplikace ke cRIO a jeden nezávislý cyklus. Stejně jako v aplikaci „RT“ i v této aplikaci si cykly předávají datový klastr se sejmutými vzorky díky funkci datové fronty, dále „Fronta“. V nezávislém cyklu je využita „Event“ struktura, která spouští jednotlivé části kódu v závislosti na události, ve všech případech je těmito událostmi myšleno stisknutí některého z ovládacích prvků. V tomto cyklu vykonávají svou funkci prvky, které k tomu nepotřebují přístup k síťovému rozhraní.

První časovaný cyklus závislý na připojení k aplikaci cRIO obstarává posílání příkazu aplikaci „RT“. Délka cyklu je nastavena 100ms. V závislosti na zvoleném ovládacím prvku, cyklus odesílá zvolené příkazy aplikaci „RT“. V případě ovládacího prvku „Konfigurace“ dojde k zaslání konfiguračního klastru, který je tvořen hodnotami ovládacích prvků nacházejících se v kapitole 6.4.2.1 Okno Konfigurace.

Druhý časovaný cyklus obstarává přijímání dat z aplikace cRIO. Cyklus se opakuje každou jednu milisekundu. Přijaté zprávy se třídí na základě přijatých Meta datech. Přijímají se tři druhy zpráv: „Kalibrace“, „Stav“, „Data“. Při vytvoření spojení dojde k přijetí klastru obsahujícího kalibrační hodnoty vstupů analogového modulu NI 9215. Přijmutím zprávy „Stav“ posílá aplikace „RT“ hodnoty o důležitých indikačních prvcích, kterými informuje o svém stavu. Zprávy typu „Data“ obsahují datový klastr obsahující pole sejmutých vzorků a časovou známku. Tento klastr je poté uložen v datové Frontě.

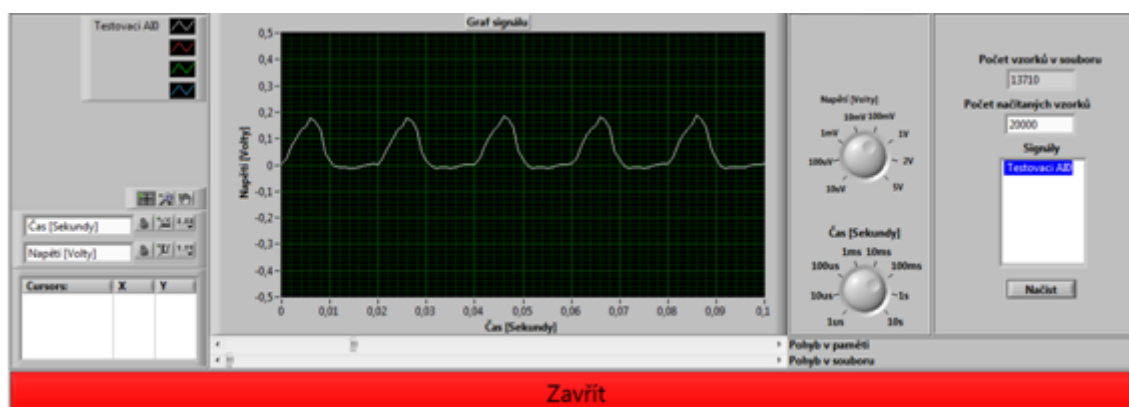
Za podmínky, že se ve frontě nachází jeden nebo více prvků, obstarává třetí časovaný cyklus jejich vyjmutí a zpracování. V první části je datový cyklus rozdělen na jednotlivé prvky. Cyklus se dále dělí na část pro analogový a část pro digitální signál. Každá jednotlivá část je spuštěna pouze za předpokladu, že je snímán minimálně jeden z daného druhu signálu. Do těchto částí vstupují již vzorky jednotlivého typu signálu, časová známka a delta T. V analogové části jsou hodnoty jednotlivých vzorků převedeny na hodnotu ve Voltech za pomoci vnořené aplikace „Převod Hodnot“. Dále je vytvořena datová struktura „Waveform“ skládající se z vzorků signálu, delta T a časové známky. Vzhledem k tomu, že při vzorkovacích frekvencích nad 10k Hz dochází k problémům s vytvářením struktury waveform, je signál kontrolován funkcí „resample waveform“, která zajišťuje udělení správné časové známky. V případě zvolení doplňkových funkcí je signál zpracován filtrací nebo i matematickými funkcemi. Poté je signál prezentován na grafu a ukládán do souboru, pokud je tak zvoleno.



Obr. 20: Vývojový diagram HOST aplikace

## 6.5 Aplikace Vizualizace

Aplikace je určena k zobrazení, zpracování signálů uložených v souboru a je závislá pouze na aplikaci Host, odkud je tato aplikace spuštěna. V úvodu je uživatel dotázán na zvolení souboru pro načtení dialogovým oknem. Na výběr se zobrazují pouze soubory s příponou \*.tdms. Aplikace obsahuje graf, který zobrazuje načtený signál. Uživatel má při více kanálech možnost vybrat, které chce zobrazit. Tyto zvolené signály jsou načteny do paměti o velikosti dané počtem načítaných vzorků. Pro pohyb v souboru slouží níže položený posuvník, posuvník výše slouží pro pohyb ve vzorcích uložených v paměti.



Obr. 21: Přední panel Aplikace Vizualizace

### 6.5.1 Graf

Graf slouží pro zobrazení načtených signálů ze souboru. Graf je standardně nastaven pro zobrazení času na ose X v jednotce [s]-sekundy a napětí na ose Y [Volty].

V levém horním rohu grafu se nachází legenda signálů se zobrazením typu a barvy čáry pro každý signál. Pod legendou se nachází ovládání grafu, nastavení souřadnic a kurzory. Více popsáno v kapitole 6.6 Manipulace s grafy.

### 6.5.2 Ovládací prvky grafu

Ovládací prvky grafu jsou umístěny v pravém horním rohu grafu. Pro ovládání osy Y slouží otáčecí přepínací prvek *Y osa*. Pro nastavení rozsahu X osy slouží prvek *X osa*.

- *Y osa*

Slouží pro nastavení rozsahu Y souřadnice v jednotkách Jednotky/Dílek. Celkový rozsah osy Y je potom dán dle vztahu (1).

$$Y_{min,max} = \pm(5 * Y_{osa}) \quad [\text{Volty}] \quad (1)$$



- ***X osa***

Nastavení celkového rozsahu X souřadnice v jednotkách [s]- sekunda.

### 6.5.3 Prvky aplikace Vizualizace

- **Indikátor *Počet vzorků v souboru*** zobrazuje celkový počet vzorků signálu v souboru.
- **Nastavení *Počet načítaných vzorků*** nastavuje počet načítaných vzorků ze souboru do paměti.
- **Menu *Signály*** zobrazí a umožňuje vybírat uložené signály v souboru. Signály se vybírají pomocí stisknutí klávesy *CTRL* a kliknutím myši na daný signál. Signály jsou zobrazeny po stisknutí tlačítka *Načíst*
- **Tlačítka *Načíst*** načte vybrané signály
- **Posuvník *Pohyb v souboru*** slouží pro pohyb v souboru. Posun je možný posuvným boxem, kliknutím na koncové šipky dojde k posuvu o nastavený počet načítaných vzorků, kliknutím mezi posuvný box a šipky dochází k jemnému posuvu.
- **Posuvník *Pohyb v paměti*** slouží pro pohyb v paměti. Posun je možný posuvným boxem, kliknutím na koncové šipky a mezi posuvný box a šipky dochází k jemnému posuvu.

## 6.6 Manipulace s grafy

### 6.6.1 Manipulace s průběhem grafu

Prvky v tomto bloku slouží k manipulaci se zobrazenými průběhy (změna měřítek, zoom, posun kursorů). Manipulace se zobrazeným průběhem pomocí těchto prvků nemá vliv na velikost úseku průběhu dat v paměti.

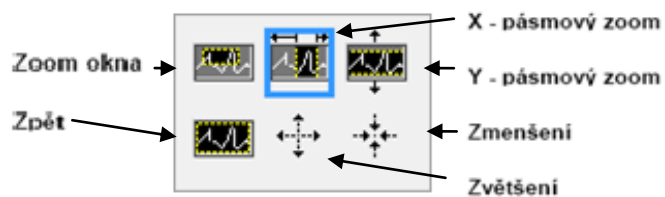


Obr. 22: Panel ovládání průběhu grafu

**Aktivace kursoru** - přepne ukazatel myši v grafu na posun kursoru

**Posun grafu** - přepne ukazatel myši v grafu na posun celého průběhu

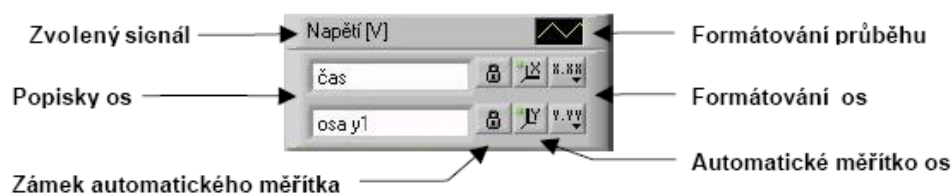
**Zoom grafu** - přepne ukazatel myši v grafu na zoom (vybraný typ)



Obr. 23: Panel zoom

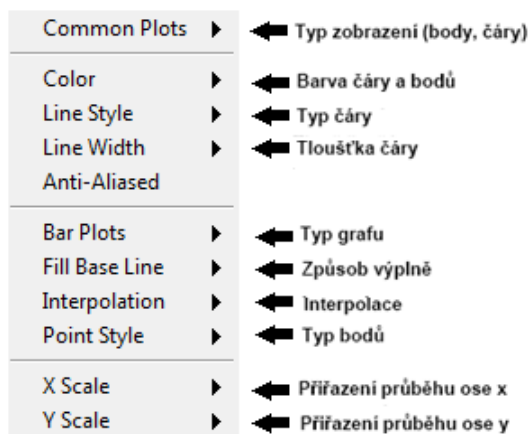
### 6.6.2 Legenda grafu a formátování grafu

V oblasti Legenda grafu jsou obsaženy ovládací prvky pro formátování jednotlivých průběhů, souřadnicových os, pro přiřazení osy Y průběhům, nastavení automatického měřítka a podobně.



Obr. 24: Legenda a formátování os

**Formátování průběhu** - slouží ke změně vlastností průběhu. Kliknutím na řádek kanálu se rozbalí nabídka umožňující provádět všechny operace.



Obr. 25:Legenda a formátování os

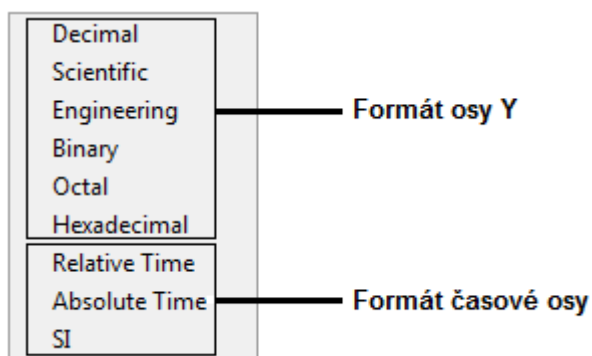
**Popisky Os** - popisky osy X a Y. Text v tomto poli se použije u příslušné osy jako titulek.

**Automatické měřítko os** - automatické měřítko je nástroj pro maximální využití plochy grafu. Rozsah příslušné osy se automaticky přizpůsobuje měřené veličině tak, aby celý průběh zůstal uvnitř grafu. Pro vypnutí automatického měřítka je nejprve nutné je odemknout tlačítkem **Zámek automatického měřítka**.

**Formátování os** - používá se ke změnám vlastností os (formát, přesnost, barva mřížky...)

**Formát osy** - nastavení formátu osy. U os Y přichází do úvahy prvních šest možností, pro časovou osu (osa X) se vybere buď absolutní čas (obsahuje i datum kvůli možnosti déletrvajících měření), nebo čas relativní (v sekundách, začátek měření je 0.0). Po změně formátu časové osy je nutné obnovit graf tlačítkem **Načíst**.

Při manipulacích s formátem osy se doporučuje mít zapnuté automatické měřítko.



Obr. 26:Formátování os

## 6.7 Komunikace přes TCP/IP pomocí STM

Komunikační protokol by měl mít následující vlastnosti:

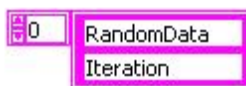
- Jednoduše balí a rozbaluje data.
- Skrývá implementační detaily transportní vrstvy (TCP/IP, UDP, atd..).
- Minimalizuje síťový provoz posíláním dat, pouze když je to potřeba.
- Minimalizuje dopad na celkové režijní náklady a propustnost.
- Umožňuje komunikovat i s jinými prostředím než Labview (C, C++, atd..)

### 6.7.1 Implementace STM protokolu v LabVIEW

STM je generický Ethernetový protokol pro LabVIEW. Instalátor pro STM protokol, který obsahuje aplikační interface a příklad kódu, je volně ke stažení na internetové stránce firmy National Instruments [www.ni.com](http://www.ni.com). V každém komunikačním protokolu jsou hlavičková data (Meta data), na kterých závisí zpracování datového toku na přijímacím konci, neboli „Data o datech“. Posíláním Meta dat v každém balíčku má za následek zvýšení datového přenosu. Při zaměření na vysoko výkonnostní aplikace je vhodné minimalizovat komunikační provoz na minimum tím, že se zašle redukovaný soubor Meta dat. STM protokol využívá postupu vytvoření samostatného balíčku Meta dat obsahující indexovaný seznam Meta informací, které jsou odesílány při vytvoření spojení mezi uživateli. Tímto dosáhneme, že každý host má všechny informace, které potřebuje k dekódování následujících zpráv. Při odesílání zprávy zakóduje odesílatel zprávu s Meta indexem, který příjemce využije k identifikaci zprávy a volby dekódovacích informací.

#### 6.7.1.1 Meta Data

Meta data jsou implementována jako pole klustrů. Každý prvek pole obsahuje informace potřebné k zabalení a dekódování jedné proměnné hodnoty. Uživatel může definovat pouze jméno, nebo může přizpůsobit Meta vlastnosti (např. datový typ) podle požadavků aplikace.



Obr. 27: Příklad Metadat dvou proměnných, Randomdata a Iteration

Před posíláním dat je vytvořen balíček obsahující velikost dat, Metadata ID (jméno Metadata) a samotná data.

Velikost Dat (32 bits)	Metadata ID (16 bits)	Data
---------------------------	--------------------------	------

Obr. 28: Formát Metadata balíčku

## 6.8 Použitý souborový systém

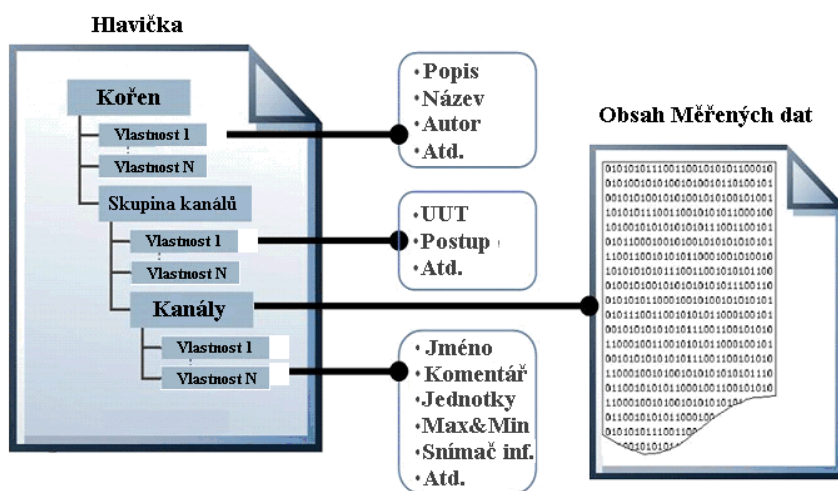
### 6.8.1 Souborový systém TDMS

Ke snížení potřeby navrhování a nakládání s vlastním datovým formátem National Instrument vytvořil flexibilní datový model TDMS, který je přirozeně přístupný v NI Labview, LabWindows/CVI, a DIAdem, a je přenosný mezi jinými aplikacemi např. Microsoft Excel. TDMS datový model nabízí různé unikátní výhody, jako schopnost nastavit podle vlastních projektových potřeb a jednoduše připojit popisové informace k měření při zapisování dat na disk. TDMS souborový systém je optimalizovaný pro dobře dokumentovaná data na disku.

Pro načtení TDMS souboru v Microsoft Excel je potřeba stáhnout z internetových stránek firmy NI jako volně dostupnou doplňkovou funkci.

### 6.8.2 Formát souborového systému TDMS

TDMS souborový systém obsahuje 3 Úrovně hierarchie, jak lze vidět na Obr. 6 – Kořen, Skupina a Kanály. Každá úroveň umožňuje nekonečný počet uživateli definovatelných atributů, tvořící soubor „Search-ready“.



Obr. 29: Každý TDMS soubor zapsaný v LabView obsahuje popisné informace o kořenu, skupinách kanálu a kanálech.

---

TDMS souborový systém navíc obsahuje binární indexový soubor s příponou \*.TDMS\_Index. Tento soubor poskytuje sloučené informace o všech vlastnostech obsahu datového souboru a zrychluje přístup k datu při čtení. TDMS\_Index je automaticky vygenerován, pokud je ztracen.

## 7 Závěr

V sedmé a zároveň v poslední kapitole bych se chtěl stručně vrátit ke všem čtyřem bodům zadání diplomové práce a vytvořit krátké shrnutí toho, o čem tato diplomová práce pojednává.

Virtuální instrumentace je opravdu široký pojem a určitě by se dalo k tomuto tématu napsat mnoho knih.

První kapitola je krátkým nahlédnutím do problematiky virtuálních přístrojů.

Druhou kapitolu jsem věnoval vývojovým prostředím virtuální instrumentace. Zaměřil jsem se převážně na prostředí NI LabView vzhledem k jeho využití ke zpracování praktické části této diplomové práce. V kapitole jsou stručně popsána vývojová prostředí ostatních výrobců.

Pro pochopení funkce dataloggeru se ve třetí kapitole věnuji obecnému popisu tohoto zařízení. Kapitola by měla pomoci k porozumění dataloggeru, jeho funkcím a vlastnostem.

Ve čtvrté kapitole popisují vlastnosti, architekturu a způsob programování technologie programovatelných hradlových polí. Vzhledem k dalšímu využití v praktické části této práce popisují v kapitole dále základní rysy platformy CompactRIO, možnost využití a parametry jednotlivých částí.

Závěrečné dvě kapitoly jsou věnovány samotné praktické části této práce. Těžiště virtuální instrumentace leží v softwarové části, proto je kapitola věnovaná realizaci softwarové části obsáhlejší. Hardwarová kapitola popisuje celé měřicí schéma a technických parametry použitých zařízení při konstruování dataloggeru. Softwarová kapitola pojednává o struktuře aplikací a rozebírá jednotlivé aplikace s funkčního i uživatelského hlediska.

Hlavní bod zadání spočíval v naprogramování virtuálního dataloggeru. Věřím, že jsem udělal při programování maximum, co se v rámci znalostí a nabytých zkušeností s programováním v programovacím jazyce G a znalostí dané problematiky udělat dalo. Pro odladění modulů jsem využil programu generátor signálu přes zvukovou kartu PCLAB verze 2.3. Při měření bylo dosaženo maximální vzorkovací frekvence 100KS/s při snímání jednoho signálu, při snímání všech signálů bylo dosaženo až 50 KS/s, ale v rámci zaručení stability programu jsem se rozhodl ji snížit na 30KS/s. Při měření jsem ověřil jednotlivé funkce dataloggeru.

---

V aplikaci dataloggeru mohlo být naprogramováno více funkcí, jako například možnost využití většího množství modulů a lepší využití jejich možností, přidání převodní veličiny pro snímání různých signálů přes převodník. Velmi zajímavé a z hlediska uživatele určitě užitečné by bylo doplnění o možnost zobrazení průběhů signálu v samostatných pop-up oknech, ale realizace této funkce ztroskotala na velké časové prodlevě mezi jednotlivými cykly. Pro rozšíření funkcí by bylo potřeba více času a více zkušeností s vývojovým prostředím.



## Použitá literatura

- [1] Židek, J.: Základní principy virtuální instrumentace, učební texty VŠB-TU, Ostrava
- [2] Kaminský, D.: Parametry převodníkových karet, učební texty VŠB-TU, Ostrava
- [3] Židek, J.: Úvod do vývojového prostředí LabVIEW, učební texty VŠB-TU, Ostrava
- [4] Vetešník, R.: Návrh a realizace základních měřicích systémů ve vývojovém prostředí LabVIEW, diplomová práce, VŠB-TU, Ostrava 1997
- [5] Datasheet cRIO 9073 , National Instruments (Přiloženo na DVD)  
URL: <http://sine.ni.com/ds/app/doc/p/id/ds-204/lang/en>
- [6] Datasheet modulu NI 9215 , National Instruments (Přiloženo na DVD)  
URL: <http://sine.ni.com/nips/cds/print/p/lang/en/nid/208814>
- [7] Datasheet modulu NI 9422 , National Instruments (Přiloženo na DVD)  
URL: <http://sine.ni.com/nips/cds/print/p/lang/en/nid/208814>
- Ďaďo, S.: Syntéza měřicích přístrojů, učební texty ČVUT, Praha

---

## Seznam příloh

Příloha č. 1      Zapojení vstupů Modulů

Příloha č. 2      Čelní panel Virtuálního Dataloggeru

Příloha č. 3      Bloková schéma Virtuálního Dataloggeru